

JP041015335

12.10.2004

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

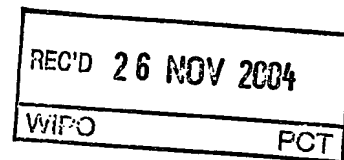
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2003年10月10日

出願番号
Application Number: 特願2003-352913

[ST. 10/C]: [JP2003-352913]

出願人
Applicant(s): 松下電器産業株式会社

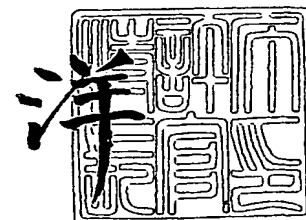


PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2004年11月12日

特許庁長官
Commissioner,
Japan Patent Office

小川



BEST AVAILABLE COPY

【書類名】 特許願
【整理番号】 2131150541
【提出日】 平成15年10月10日
【あて先】 特許庁長官 殿
【国際特許分類】 G11B 7/00
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 池田 航
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 岩本 啓明
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 岡田 智之
【特許出願人】
 【識別番号】 000005821
 【氏名又は名称】 松下電器産業株式会社
【代理人】
 【識別番号】 100090446
 【弁理士】
 【氏名又は名称】 中島 司朗
【手数料の表示】
 【予納台帳番号】 014823
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9003742

【書類名】 特許請求の範囲**【請求項 1】**

映像データとプログラムが記録されており、
映像データの管理区間毎に、前記プログラムの状態を制御する管理情報が記録されてい
る

ことを特徴とする情報記録媒体。

【請求項 2】

前記管理区間とは、前記映像データをシナリオごとに分割したタイトル、情報記録媒体
全体、複数の情報記録媒体の何れかである

ことを特徴とする請求項 1 記載の情報記録媒体。

【請求項 3】

前記管理区間とともに、前記管理区間の一部を指定できる前記管理情報が記録されてい
る請求項 2 記載の情報記録媒体。

【請求項 4】

前記管理区間ごとの制御とは、前記管理区間終了時に前記プログラムを終了させること
である、請求項 1 記載の情報記録媒体。

【請求項 5】

前記管理区間ごとの制御とは、前記管理区間開始時に前記プログラムを起動させること
である、請求項 1 記載の情報記録媒体。

【請求項 6】

前記管理情報は、前記管理区間開始時に前記プログラムを起動するかしないか識別する
ための情報を有する

ことを特徴とする請求項 1 記載の情報記録媒体。

【請求項 7】

前記管理情報が、前記管理区間開始時に前記プログラムを一時停止するか再開するか識
別するための情報を有する

ことを特徴とする請求項 1 記載の情報記録媒体。

【請求項 8】

2 つ以上の前記管理情報には、同じプログラムに関する管理情報が含まれており、
前記管理情報は、一の管理区間から他の管理区間への遷移を制御する情報であり、
当該管理情報により制御される遷移とは、
プログラムの終了及び再起動を介さず、遷移前の管理区間でのプログラムの状態を遷移
後の管理区間に継続させることである、請求項 1 記載の情報記録媒体。

【請求項 9】

前記プログラムとはイベントを受けて駆動するイベントハンドラを備えていることを特
徴とする請求項 1 記載の情報記録媒体。

【請求項 10】

前記イベントとはユーザー操作がなされたことを示すイベント、

前記映像データの所定の時点に達したことを示すイベント、

機器の状態が変化したことを示すイベント

の何れかであることを特徴とする請求項 9 記載の情報記録媒体。

【請求項 11】

前記イベントとは管理単位の終了を示すイベントであり、イベントに対応したプロセス
がプログラムを終了させる

ことを特徴とした請求項 10 記載の情報記録媒体。

【請求項 12】

前記イベントとは管理単位の開始を示すイベントと終了を示すイベントであり、

終了を示すイベントに対応したプロセスがプログラムを一時停止させ、開始を示すイ
ベントに対応したプロセスがプログラムを再開させる

ことを特徴とした請求項 10 記載の情報記録媒体。

【請求項 1 3】

映像データとプログラム、及び、前記プログラムが参照するデータが記録されており、管理区間開始時に前記プログラムをメモリ上に読み込み、前記管理区間終了時にメモリ上に読み込まれた前記プログラムを解放するための管理情報が記録されていることを特徴とする情報記録媒体。

【請求項 1 4】

前記管理区間とは、前記映像データをシナリオごとに分割したタイトル、情報記録媒体全体、複数の情報記録媒体の何れかであることを特徴とする請求項 1 3 記載の情報記録媒体。

【請求項 1 5】

前記管理区間とともに、前記管理区間の一部を指定できる前記管理情報が記録されている、請求項 1 4 記載の情報記録媒体。

【請求項 1 6】

2 つ以上の前記管理情報は、同じプログラム及び前記プログラムが参照するデータに関する管理情報であり、前記管理情報は、一の管理区間から他の管理区間への遷移を制御する情報であり、当該管理情報により制御される遷移とは、プログラムの終了及び再起動、並びに、データの解放及び読み込みを介さず、遷移前の管理区間でのプログラム及びデータを保持したまま遷移後の管理区間に継続させることである、請求項 1 記載の情報記録媒体。

【請求項 1 7】

前記管理情報はメモリ上にデータ読み込むための優先度情報を備える、請求項 1 3 記載の情報記録媒体。

【請求項 1 8】

前記優先度情報とは、メモリに必ず読み込まなければならないデータと必ずしも読み込まなくてもよいデータとを区別することを特徴とする請求項 1 8 記載の情報記録媒体。

【請求項 1 9】

前記管理情報とは情報記録媒体上のファイルとメモリ上のファイル情報を備えていることを特徴とする請求項 1 3 記載の情報記録媒体。

【請求項 2 0】

前記管理情報は、メモリ上の複数のファイルを情報記録媒体上の 1 つのファイルとして対応づけて管理できることを特徴とする請求項 1 9 記載の情報記録媒体。

【請求項 2 1】

請求項 1 記載の管理情報と請求項 1 3 記載の管理情報において、前記管理区間が同じ区間である管理データを結合することにより、プログラムの状態とデータの読み込みを制御することが可能な管理情報を備えたことを特徴とする情報記録媒体。

【請求項 2 2】

請求項 2 1 記載の管理情報が請求項 6 又は 7 記載の識別情報を備えており、データ読み込みの優先度は前記識別情報に対応して一意に決めることができることを特徴とする情報記録媒体。

【請求項 2 3】

前記プログラムが前記映像データとは別ファイルとして、別の場所に記録されていることを特徴とした請求項 1 記載の情報記録媒体。

【請求項 2 4】

前記プログラムが前記映像データとともに多重化されていることを特徴とした請求項 1 記載の情報記録媒体。

【請求項 2 5】

前記プログラムが前記映像データの途中にインターリーブ配置されて記録されていることを特徴とした請求項 1 記載の情報記録媒体。

【請求項 26】

前記管理情報による管理及び制御は、
請求項 23～請求項 25 の何れかに記載のプログラムを対象としている
ことを特徴とする、請求項 21 記載の情報記録媒体。

【請求項 27】

請求項 21 記載の管理情報において、同じ前記管理単位内で同時に起動されないプログラム同士をグループ化して管理する
ことを特徴とする情報記録媒体。

【請求項 28】

前記管理情報が映像データとは別に管理されている
ことを特徴とする請求項 1, 13, 21 の何れかに記載の情報記録媒体。

【請求項 29】

前記管理情報が映像データと同じファイルとして管理されている
ことを特徴とする請求項 1, 13, 21 の何れかに記載の情報記録媒体。

【請求項 30】

前記プログラムの管理情報とは別に、前記映像データの再生制御情報が記録されている
ことを特徴とした情報記録媒体。

【請求項 31】

映像データとプログラムが記録された情報記録媒体を再生する再生装置であって、
前記映像データを再生する再生部と、
前記プログラムを処理するプログラム処理部と、
管理情報にしたがって管理区間ごとに前記プログラムの状態を制御する手段と を備える
ことを特徴とする再生装置。

【請求項 32】

前記プログラムの状態を制御する手段とは、プログラムの起動あるいはプログラムの終了、あるいはプログラムの一時停止と再開の何れかである
ことを特徴とする再生装置。

【請求項 33】

前記管理情報とは、請求項 1 記載の管理情報である
ことを特徴とする請求項 31 記載の再生装置。

【請求項 34】

映像データとプログラムが記録された情報記録媒体を再生する再生装置であって、
前記プログラムをメモリに読み込む手段とメモリから解放する手段を備え、管理情報にしたがって管理区間ごとにデータを制御する
ことを特徴とする再生装置。

【請求項 35】

前記管理情報とは、請求項 13 記載の管理情報であることを特徴とする請求項 34 記載の再生装置。

【請求項 36】

請求項 31 記載のプログラムの状態を制御する手段とは、
管理区間開始時にプログラムを起動する手段、
管理区間終了時にプログラムを終了する手段、
イベントによりプログラムを起動する手段、
イベントによりプログラムを終了する手段、
プログラムからの通知により別のプログラムを起動する手段、
プログラムからの通知により別のプログラムを終了する手段
の何れか、あるいは、いくつかの手段であることを特徴とする再生装置。

【請求項 37】

前記プログラムに終了通知を行っても前記プログラムが自動的に終了しない場合、前記プログラムの状態を制御する手段がプログラムを強制的に終了させる手段

を備えることを特徴とする請求項 31 記載の再生装置。

【請求項 38】

前記メモリ上のプログラムをプログラム処理部に読み込む手段とプログラム処理部からプログラムを解放する手段

を備えたことを特徴とする請求項 34 記載の再生装置。

【請求項 39】

請求項 34 記載の再生装置であって、前記メモリにプログラムを読み込めるか否かを、プログラムのサイズとメモリの空き容量を比較して判定する手段と、読み込めると判定した場合に一つ又は複数のプログラムを選択してメモリに読み込む手段を備える

ことを特徴とする再生装置。

【請求項 40】

前記判定手段として、請求項 17 記載の優先度情報の順で前記プログラムのサイズとメモリの空き容量を比較を行う

ことを特徴とする請求項 39 記載の再生装置。

【請求項 41】

請求項 38 記載の再生装置において、前記メモリ内にデータが読み込まれていないため、前記プログラム処理部にプログラムを読み込めない場合、データの読み込みに失敗する手段、

タイマーを設定して指定時間まで読み込み待ちを行い、指定時間が過ぎてもデータが読み込めないときに読み込みに失敗する手段、

データが読み込めるまで待ち続ける手段

の何れかを備えたことを特徴とする再生装置。

【請求項 42】

請求項 34 記載の再生装置であり、前記映像データを読み込み中であつたも、前記プログラムを前記メモリに読み込む手段を

備えたことを特徴とする再生装置。

【請求項 43】

前記映像データは再生制御部で制御され、

再生制御部は、

前記プログラム処理部による処理にて前記プログラムが異常状態になった場合でも、前記映像データの再生を保証する

ことを特徴とする請求項 31 記載の再生装置。

【請求項 44】

請求項 17 記載の優先度情報により前記プログラムの読み込みを判定する方法であつて

優先度を判定し優先度順に処理を行うステップと、

前記プログラムのサイズとメモリの空き容量を比較するステップと、

前記メモリの空き容量が前記プログラムサイズより大きい場合は前記プログラムを読み込むステップと

を備えることを特徴とする再生方法。

【書類名】明細書

【発明の名称】情報記録媒体、その再生装置及び再生方法

【技術分野】

【0001】

本発明は、BD-ROM等、映像データを記録した情報記録媒体、その再生装置及び再生方法に関し、特に、映像データとプログラムとを含むコンテンツの開発にプログラミング環境を導入する場合の改良に関する。

【背景技術】

【0002】

映像データを記録した情報記録媒体の代表格は、DVD（以下、SD-DVDまたは単にDVDと称する）である。以降、従来のDVDについて説明する。

図1は、SD-DVDの構造を示した図である。図1の下段に示すように、DVDディスク上にはリードインからリードアウトまでの間に論理アドレス空間が設けられ、論理アドレス空間の先頭からファイルシステムのボリューム情報が記録され、続いて映像音声などのアプリケーションデータが記録されている。

【0003】

ファイルシステムとは、ISO9660やUDF（Universal Disc Format）のことであり、ディスク上のデータをディレクトリまたはファイルと呼ばれる単位で表現する仕組みである。日常使っているPC（パーソナルコンピュータ）の場合でも、FATまたはNTFSと呼ばれるファイルシステムを通すことにより、ディレクトリやファイルという構造でハードディスクに記録されたデータがコンピュータ上で表現され、ユーザビリティを高めている。SD-DVDの場合、UDF及びISO9660両方を使用しており（両方を合わせて「UDFブリッジ」と呼ぶ事がある）、UDFまたはISO9660どちらのファイルシステムドライバによってもデータの読み出し（ここで取り扱うDVDはパッケージメディア用のROMディスクであり、物理的に書き込みが不可能である）ができるようになっている。

【0004】

DVD上に記録されたデータは、UDFブリッジを通して、図1左上に示すようなディレクトリまたはファイルとして見ることができる。ルートディレクトリ（図中「ROOT」）の直下に「VIDEO_TS」と呼ばれるディレクトリが置かれ、ここにDVDのアプリケーションデータが記録されている。アプリケーションデータは、複数のファイルとして記録され、主なファイルとして以下のものがある。

VIDEO_TS.IFO	ディスク再生制御情報ファイル
VTSC_01_0.IFO	ビデオタイトルセット#1再生制御情報ファイル
VTSC_01_0.VOB	ビデオタイトルセット#1ストリームファイル
.....	

拡張子として2つの種類が存在する。「IFO」は再生制御情報が記録されたファイルであって、「VOB」はAVデータであるMPEGストリームが記録されたファイルである。再生制御情報とは、DVDで採用されたインタラクティビティ（ユーザの操作に応じて再生を動的に変化させる技術）を実現するための情報や、メタデータのようなタイトルやAVストリームに付属する情報などのことである。また、DVDでは一般的に再生制御情報のことをナビゲーション情報と呼ぶことがある。

【0005】

再生制御情報ファイルは、ディスク全体を管理する「VIDEO_TS.IFO」と、個々のビデオタイトルセット（DVDでは複数のタイトル、言い換えれば異なる映画や異なるバージョンの映画を1枚のディスクに記録することが可能である。）毎の再生制御情報である「VTSC_01_0.IFO」がある。ここで、ファイル名ボディにある「01」はビデオタイトルセットの番号を示しており、例えば、ビデオタイトルセット#2の場合

合は、「VTS_02_0. IFO」となる。

【0006】

図1の右上部は、DVDのアプリケーション層でのDVDナビゲーション空間であり、前述した再生制御情報が展開された論理構造空間である。「VIDEO_TS. IFO」内の情報は、VMGI (Video Manager Information) として、「VTS_01_0. IFO」または、他のビデオタイトルセット毎に存在する再生制御情報はVTSI (Video Title Set Information) としてDVDナビゲーション空間に展開される。

【0007】

VTSIの中にはPGC (Program Chain) と呼ばれる再生シーケンスの情報であるPGCI (Program Chain Information) が記述されている。PGCIは、Cellの集合とコマンドと呼ばれる一種のプログラミング情報によって構成されている。Cell自身はVOB (Video Objectの略であり、MPEGストリームを指す) の一部区間または全部区間の集合であり、Cellの再生は、当該VOBのCellによって指定された区間を再生することを意味している。

【0008】

コマンドは、DVDの仮想マシンによって処理されるものであり、ブラウザ上で実行されるJava (登録商標) スクリプトなどに近いものである。しかしながらJava (登録商標) スクリプトが論理演算の他にウィンドウやブラウザの制御 (例えば、新しいブラウザのウィンドウを開くなど) を行うのに対して、DVDのコマンドは、論理演算の他にAVタイトルの再生制御、例えば、再生するチャプタの指定などを実行するだけのものである点で異なっている。

【0009】

Cellはディスク上に記録されているVOBの開始及び終了アドレス (論理アドレス) をその内部情報として有しており、プレーヤは、Cellに記述されたVOBの開始及び終了アドレス情報を使ってデータの読み出し、再生を実行する。

図1はAVストリーム中に埋め込まれているナビゲーション情報を説明する概略図である。SD-DVDの特長であるインタラクティビティは前述した「VIDEO_TS. IFO」や「VTS_01_0. IFO」などに記録されているナビゲーション情報だけではなく、幾つかの重要な情報はナビゲーション・パック (ナビパックまたは、NV_PCKと称する) と呼ばれる専用キャリアを使いVOB内に映像、音声データと一緒に多重化されている。

【0010】

ここでは簡単なインタラクティビティの例としてメニューを説明する。メニュー画面上には、幾つかのボタンが現れ、夫々のボタンには当該ボタンが選択実行された時の処理が定義されている。また、メニュー上では一つのボタンが選択されており (ハイライトによって選択ボタン上に半透明色がオーバーレイされている)、ユーザは、リモコンの上下左右キーを使って、選択状態のボタンを上下左右の何れかのボタンに移動させることが出来る。リモコンの上下左右キーを使って、選択実行したいボタンまでハイライトを移動させ、決定する (決定キーを押す) ことによって対応するコマンドのプログラムが実行される。一般的には対応するタイトルやチャプタの再生がコマンドによって実行されている。

【0011】

図2の左上部はNV_PCK内の概要を示している。

NV_PCK内には、ハイライトカラー情報と個々のボタン情報などが含まれている。ハイライトカラー情報には、カラーパレット情報が記述され、オーバーレイ表示されるハイライトの半透明色が指定される。ボタン情報には、個々のボタンの位置情報である矩形領域情報と、当該ボタンから他のボタンへの移動情報 (ユーザの上下左右キー操作夫々に対応する移動先ボタンの指定) と、ボタンコマンド情報 (当該ボタンが決定された時に実行されるコマンド) が記述されている。

【0012】

メニュー上のハイライトは、図2の中央右上部に示すように、オーバーレイ画像として作られる。オーバーレイ画像は、ボタン情報の矩形領域情報にカラーパレット情報の色をつけた物である。このオーバーレイ画像は図46の右部に示す背景画像と合成されて画面上に表示される。

上述のようにして、DVDではメニューを実現している。また、何故、ナビゲーションデータの一部をNV_PCKを使ってストリーム中に埋め込んでいるのは、ストリームと同期して動的にメニュー情報を更新、例えば、映画再生中の途中5分～10分の間にだけメニューが表示されるなど、同期タイミングが問題となりやすいアプリケーションの場合でも、問題なく実現できるようにしたためである。

【0013】

図3は、DVDのVOBのイメージである。図に示すように、映像、音声、字幕などのデータ(A段)は、MPEGシステム(ISO/IEC13818-1)規格に基づいて、パケット及びパック化し(B段)、夫々を多重化して1本のMPEGプログラムストリームにしている(C段)。また、前述した通りインタラクティブを実現するためのボタンコマンドを含んだNV_PCKも一緒に多重化をされている。

【0014】

MPEGシステムの多重化の特徴は、多重化する個々のデータは、そのデコード順に基づくビット列になっているが、多重化されるデータ間、即ち、映像、音声、字幕の間は必ずしも再生順、言い換えればデコード順に基づいてビット列が形成されているわけではない。これはMPEGシステムストリームのデコーダモデル(一般にSystem Target Decoder、またはSTDと呼ばれる(図3のD段))が多重化を解いた後に個々のエレメンタリーストリームに対応するデコーダバッファを持ち、デコードタイミングまでに一時的にデータを蓄積している事に由来している。このデコーダバッファは、個々のエレメンタリーストリーム毎にサイズが異なり、映像に対しては、232kB、音声に対しては4kB、字幕に対しては52kBを夫々有している。このため、各デコーダバッファへのデータ入力タイミングは個々のエレメンタリーストリームで異なるため、MPEGシステムストリームとしてビット列を形成する順番と表示(デコード)されるタイミングにずれが生じている。

【0015】

即ち、映像データと並んで多重化されている字幕データが必ずしも同一タイミングでデコードされているわけではない。

かかるDVDの構成は、以下の特許文献1に記載されている。

【特許文献1】特許第2813245号

【発明の開示】

【発明が解決しようとする課題】

【0016】

ところで映像データと、プログラムとを含むコンテンツを、情報記録媒体で配布しようとすると、プログラムの状態がおかしくなることによりプレーヤ自体の動作やAV再生が破綻してしまふ恐れがある。次世代の情報記録媒体であるBD-ROMは、上述したようなナビゲーションコマンドではなく、Java(登録商標)プログラムを含むコンテンツの記録が予定されており、かかるコンテンツにあたっては、プレーヤの動作保障が深刻な問題になる。

【0017】

メモリなどのリソースを厳密に管理するため、従来DVDプレーヤ規格では、リソースを管理する際にメモリリソースのサイズを固定化している。しかしメモリサイズの固定化は、プレーヤ設計の自由度を失わせるものであり、アプリケーションが発展してもプレーヤは、その発展から取り残されてしまうという問題点がある。

本発明の目的は、プレーヤの動作保障を実現しつつも、プレーヤ設計の自由度を高めることができる情報記録媒体を提供することである。

【課題を解決するための手段】

【0018】

上記課題を解決するため本発明にかかる情報記録媒体は、映像データとプログラムが記録されており、管理区間ごとに前記プログラムの状態を制御する管理情報が記録されていることを特徴としている。

【発明の効果】

【0019】

上述した構成では、タイトルなどを管理単位としてアプリケーションの起動及び終了やデータの読み込み及び解放のタイミングが管理可能になるので、管理単位内で動作するアプリケーションを容易に管理でき、管理単位外に影響を及ぼさない仕組みを提供することが可能となる。

ここで前記管理情報はメモリ上にデータ読み込むための優先度情報を備えていてもよい。管理情報としてデータ読み込み優先度を用いることにより、プレーヤに搭載されているメモリサイズに応じて読み込むデータサイズを調整することが可能となり、メモリサイズが小さい場合でもアプリケーションは動作保証され、メモリサイズが大きい場合はアプリケーションの動作や切替が早くなる利便性を提供することが可能となる。

【発明を実施するための最良の形態】

【0020】

(実施例1)

まず最初に本発明の第1の実施の形態について説明する。

(ディスク上の論理データ構造)

図4は、BD-ROM（以降、「BD」と称する場合もある）の構成、特にディスク媒体であるBDディスク（104）と、ディスクに記録されているデータ（101、102、103）の構成を示す図である。BDディスク（104）に記録されるデータは、AVデータ（103）と、AVデータに関する管理情報及びAV再生シーケンスなどのBD管理情報（102）と、インタラクティブを実現するBD再生プログラム（101）である。本実施の形態では、映画などのAVコンテンツを再生するためのAVアプリケーションを主眼においてのBDディスクの説明を行うが、BDディスクをCD-ROMやDVD-ROMの様にコンピュータ用途の記録媒体としてしようすることも当然のことながら可能である。図5は、上述したBDディスクに記録されている論理データを示した図である。BDディスクは、他の光ディスク、例えばDVDやCDなどと同様にその内周から外周に向けてらせん状に記録領域を持ち、内周のリード・インと外周のリード・アウトの間に論理データを記録できる論理アドレス空間を有している。また、リード・インの内側にはBCA（Burst Cutting Area）と呼ばれるドライブでしか読み出せない特別な領域がある。この領域はアプリケーションから読み出せないため、例えば著作権保護技術などに利用されることがよくある。

【0021】

論理アドレス空間には、ファイルシステム情報（ボリューム）を先頭に映像データなどのアプリケーションデータが記録されている。ファイルシステムとは従来技術で説明した通り、UDFやISO9660などのことであり、通常のPCと同じように記録されている論理データをディレクトリ、ファイル構造を使って読み出しする事が可能になっている。

【0022】

本実施例の場合、BDディスク上のディレクトリ、ファイル構造は、ルートディレクトリ（ROOT）直下にBDVIDEOディレクトリが置かれている。このディレクトリはBD-ROMで扱うAVコンテンツや管理情報などのデータ（図4で説明した101、102、103）が記録されているディレクトリである。

BDVIDEOディレクトリの下には、次の7種類のファイルが記録されている。

BD.INFO（ファイル名固定）

「BD管理情報」の一つであり、BDディスク全体に関する情報を記録したファイルである。BDプレーヤは最初にこのファイルを読み出す。

BD. PROG (ファイル名固定)

「BD再生プログラム」の一つであり、BDディスク全体に関わるプログラムを記録したファイルである。

XXX. PL (「XXX」は可変、拡張子「PL」は固定)

「BD管理情報」の一つであり、シナリオを記録するプレイリスト (Play List) 情報を記録したファイルである。プレイリスト毎に1つのファイルを持っている。

XXX. PROG (「XXX」は可変、拡張子「PL」は固定)

「BD再生プログラム」の一つであり、前述したプレイリスト毎のプログラムを記録したファイルである。プレイリストとの対応はファイルボディ名 (「XXX」が一致する) によって識別される。

YYY. VOB (「YYY」は可変、拡張子「VOB」は固定)

「AVデータ」の一つであり、VOB (従来例で説明したVOBと同じ) を記録したファイルである。VOB毎に1つのファイルを持っている。

YYY. VOBI (「YYY」は可変、拡張子「VOBI」は固定)

「BD管理情報」の一つであり、AVデータであるVOBに関わる管理情報を記録したファイルである。VOBとの対応はファイルボディ名 (「YYY」が一致する) によって識別される。

ZZZ. PNG (「ZZZ」は可変、拡張子「PNG」は固定)

「AVデータ」の一つであり、字幕及びメニューを構成するためのイメージデータPNG (W3Cによって標準化された画像フォーマットであり「ピング」と読む) を記録したファイルである。1つのPNGイメージ毎に1つのファイルを持つ。

(プレーヤの構成)

次に、前述したBDディスクを再生するプレーヤの構成について図6及び図7を用いて説明する。

【0023】

図6は、プレーヤの大まかな機能構成を示すブロック図である。

BDディスク (201) 上のデータは、光ピックアップ (202) を通して読み出される。読み出されたデータは夫々のデータの種類に応じて専用のメモリに記録される。BD再生プログラム (「BD. PROG」または「XXX. PROG」ファイルの中身) はプログラム記録メモリ (203) に、BD管理情報 (「BD. INFO」、「XXX. PL」または「YYY. VOBI」) は管理情報記録メモリ (204) に、AVデータ (「YYY. VOB」または「ZZZ. PNG」) はAV記録メモリ (205) に夫々記録される。

【0024】

プログラム記録メモリ (203) に記録されたBD再生プログラムはプログラム処理部 (206) によって、管理情報記録メモリ (204) に記録されたBD管理情報は管理情報処理部 (207) によって、また、AV記録メモリ (205) に記録されたAVデータはプレゼンテーション処理部 (208) によって夫々処理される。

プログラム処理部 (206) は、管理情報処理部 (207) より再生するプレイリストの情報やプログラムの実行タイミングなどのイベント情報を受け取りプログラムの処理を行う。また、プログラムでは再生するプレイリストを動的に変える事が可能であり、この場合は管理情報処理部 (207) に対してプレイリストの再生命令を送ることで実現する

。プログラム処理部(206)は、ユーザからのイベント、即ちリモコンキーからのリクエストを受け、ユーザイベントに対応するプログラムがある場合は、実行処理する。

【0025】

管理情報処理部(207)は、プログラム処理部(206)の指示を受け、対応するプレイリスト及びプレイリストに対応したVOBの管理情報を解析し、プレゼンテーション処理部(208)に対象となるAVデータの再生を指示する。また、管理情報処理部(207)は、プレゼンテーション処理部(208)より基準時刻情報を受け取り、時刻情報に基づいてプレゼンテーション処理部(208)にAVデータ再生の停止指示を行い、また、プログラム処理部(206)に対してプログラム実行タイミングを示すイベントを生成する。

【0026】

プレゼンテーション処理部(208)は、映像、音声、字幕／イメージ夫々に対応するデコーダを持ち、管理情報処理部(207)からの指示に従い、AVデータのデコード及び出力を行う。映像データ及び字幕／イメージの場合は、デコード後に夫々の専用プレーン、ビデオプレーン(210)及びイメージプレーン(209)に描画され、合成処理部(211)によって映像の合成処理が行われTVなどの表示デバイスへ出力される。

【0027】

図6で示すように、BDプレーヤは図4で示したBDディスクに記録されているデータ構成に基づいた構成をとっている。

図7は前述したプレーヤ構成を詳細化したブロック図である。図7では、AV記録メモリ(205)はイメージメモリ(308)とトラックバッファ(309)に、プログラム処理部(206)はプログラムプロセッサ(302)とUOPマネージャ(303)に、管理情報処理部(207)はシナリオプロセッサ(305)とプレゼンテーションコントローラ(306)に、プレゼンテーション処理部(208)はクロック(307)、デマルチプレクサ(310)、イメージプロセッサ(311)、ビデオプロセッサ(312)とサウンドプロセッサ(313)に夫々対応／展開している。

【0028】

BDディスク(201)から読み出されたVOBデータ(MPEGストリーム)はトラックバッファ(309)に、イメージデータ(PNG)はイメージメモリ(308)に夫々記録される。デマルチプレクサ(310)がクロック(307)の時刻に基づき、トラックバッファ(309)に記録されたVOBデータを抜き出し、映像データをビデオプロセッサ(312)に音声データをサウンドプロセッサ(313)に夫々送り込む。ビデオプロセッサ(312)及びサウンドプロセッサ(313)は夫々MPEGシステム規格で定める通りに、デコーダバッファとデコーダから夫々構成されている。即ち、デマルチプレクサ(310)から送りこまれる映像、音声夫々のデータは、夫々のデコーダバッファに一時的に記録され、クロック(307)に従い個々のデコーダでデコード処理される。

【0029】

イメージメモリ(308)に記録されたPNGは、次の2つの処理方法がある。イメージデータが字幕用の場合は、プレゼンテーションコントローラ(306)によってデコードタイミングが指示される。クロック(307)からの時刻情報をシナリオプロセッサ(305)が一旦受け、適切な字幕表示が行えるように、字幕表示時刻(開始及び終了)になればプレゼンテーションコントローラ(306)に対して字幕の表示、非表示の指示を出す。プレゼンテーションコントローラ(306)からデコード／表示の指示を受けたイメージプロセッサ(311)は対応するPNGデータをイメージメモリ(308)から抜き出し、デコードし、イメージプレーン(314)に描画する。

【0030】

次に、イメージデータがメニュー用の場合は、プログラムプロセッサ(302)によってデコードタイミングが指示される。プログラムプロセッサ(302)が何時イメージのデコードを指示するかは、プログラムプロセッサ(302)が処理しているBDプログラ

ムに因るものであって一概には決まらない。

イメージデータ及び映像データは、図6で説明したように夫々デコード後にイメージプレーン(314)、ビデオプレーン(315)に記録され、合成処理部(316)によって合成出力される。

【0031】

BDディスク(201)から読み出された管理情報(シナリオ、AV管理情報)は、管理情報記録メモリ(304)に記録されるが、シナリオ情報(「BD. INFO」及び「XXX. PL」)はシナリオプロセッサ(305)によって読み出され処理される。また、AV管理情報(「YYY. VOB I」)はプレゼンテーションコントローラ(306)によって読み出され処理される。

【0032】

シナリオプロセッサ(305)は、プレイリストの情報を解析し、プレイリストによって参照されているVOBとその再生位置をプレゼンテーションコントローラ(306)に指示し、プレゼンテーションコントローラ(306)は対象となるVOBの管理情報(「YYY. VOB I」)を解析して、対象となるVOBを読み出すようにドライブコントローラ(317)に指示を出す。

【0033】

ドライブコントローラ(317)はプレゼンテーションコントローラ(306)の指示に従い、光ピックアップを移動させ、対象となるAVデータの読み出しを行う。読み出されたAVデータは、前述したようにイメージメモリ(308)またはトラックバッファ(309)に記録される。

また、シナリオプロセッサ(305)は、クロック(307)の時刻を監視し、管理情報で設定されているタイミングでイベントをプログラムプロセッサ(302)に投げる。

【0034】

プログラム記録メモリ(301)に記録されたBDプログラム(「BD. PROG」または「XXX. PROG」)は、プログラムプロセッサ302によって実行処理される。プログラムプロセッサ(302)がBDプログラムを処理するのは、シナリオプロセッサ(305)からイベントが送られてきた場合か、UOPマネージャ(303)からイベントが送られてきた場合である。UOPマネージャ(303)は、ユーザからリモコンキーによってリクエストが送られてきた場合に、プログラムプロセッサ(302)にイベントを生成する。

(アプリケーション空間)

図8は、BD-ROMのアプリケーション空間を示す図である。

【0035】

BD-ROMのアプリケーション空間では、プレイリスト(Play List)が一つの再生単位になっている。プレイリストはセル(Cell)の再生シーケンスから構成される静的なシナリオと、プログラムによって記述される動的なシナリオを有している。プログラムによる動的なシナリオが無い限り、プレイリストは個々のセルを順に再生するだけであり、また、全てのセルの再生を終了した時点でプレイリストの再生は終了する。一方で、プログラムは、プレイリストを超えての再生記述や、ユーザ選択またはプレーヤの状態によって再生する対象を動的に変えることが可能である。典型的な例としてはメニューがあげられる。BD-ROMの場合、メニューとはユーザの選択によって再生するシナリオ、即ちプレイリストを動的に選択することである。

【0036】

ここで言うプログラムは、時間イベントまたはユーザイベントによって実行されるイベントハンドラの事である。

時間イベントは、プレイリスト中に埋め込まれた時刻情報に基づいて生成されるイベントである。図7で説明したシナリオプロセッサ(305)からプログラムプロセッサ(302)に送られるイベントがこれに相当する。時間イベントが発行されると、プログラム

プロセッサ (302) は ID によって対応付けられるイベントハンドラを実行処理する。前述した通り、実行されるプログラムが他のプレイリストの再生を指示することが可能であり、この場合には、現在再生されているプレイリストの再生は中止され、指定されたプレイリストの再生へと遷移する。

【0037】

ユーザイベントは、ユーザのリモコンキー操作によって生成されるイベントである。ユーザイベントは大きく2つのタイプに分けられる。一つ目は、カーソルキー（「上」「下」「左」「右」キー）または「決定」キーの操作によって生成されるメニュー選択のイベントである。メニュー選択のイベントに対応するイベントハンドラはプレイリスト内の限られた期間でのみ有効であり（プレイリストの情報として、個々のイベントハンドラの有効期間が設定されている）、リモコンの「上」「下」「左」「右」キーまたは「決定」キーが押された時に有効なイベントハンドラを検索して、有効なイベントハンドラがある場合は当該イベントハンドラが実行処理される。他の場合は、メニュー選択のイベントは無視されることになる。

【0038】

二つ目のユーザイベントは、「メニュー」キーの操作によって生成されるメニュー呼び出しのイベントである。メニュー呼び出しのイベントが生成されると、グローバルイベントハンドラが呼ばれる。グローバルイベントハンドラはプレイリストに依存せず、常に有効なイベントハンドラである。この機能を使うことにより、DVDのメニューコール（タイトル再生中に音声、字幕メニューなどを呼び出し、音声または字幕を変更後に中断した地点からのタイトル再生を実行する）を実装することができる。

【0039】

プレイリストで静的シナリオを構成する単位であるセル (Cell) はVOB (MPEGストリーム) の全部または一部の再生区間を参照したものである。セルはVOB内の再生区間を開始、終了時刻の情報として持っている。個々のVOBと一対になっているVOB管理情報 (VOBI) は、その内部にタイムマップ (Time Map または TM) を有しており、このタイムマップによって前述したVOBの再生、終了時刻をVOB内（即ち対象となるファイル「YYY.VOB」内）での読み出し開始アドレス及び終了アドレスを導き出すことが可能である。なおタイムマップの詳細は後述する。

(VOBの詳細)

図9は、本実施例で使用するMPEGストリーム (VOB) の構成図である。図9に示すように、VOBは複数のVOBU (Video Object Unit) によって構成されている。VOBUは、MPEGビデオストリームで言うGOP (Group Of Pictures) を基準として、音声データも含んだ多重化ストリームとしての一再生単位である。VOBUは0.4秒から1.0秒の時間を持ち、通常は0.5秒の再生時間を持っている。これはMPEGのGOPの構造が通常は15フレーム/秒 (NTSCの場合) によって導かれるものである。

【0040】

VOBUは、その内部にビデオパック (V_PCK) とオーディオパック (A_PCK) を有している。各パックは1セクタ、本実施例の場合は2kB単位で構成されている。

図10は、パックの構成を示した図である。

図10に示すように、ビデオデータ及びオーディオデータといったエレメンタリデータは、ペイロードと呼ばれるパケットのデータ格納領域に先頭から順次入れられていく。ペイロードにはパケットヘッダが付けられ1つのパケットを構成する。パケットヘッダには、ペイロードに格納してあるデータがどのストリームなのか、ビデオなのかオーディオなのか、また、ビデオまたはオーディオが夫々複数ストリームある場合は、どのストリームのデータなのかを識別するためのID (stream_id) と、当該ペイロードのデコード及び表示時刻情報であるタイムスタンプDTS及びPTSが夫々記録されている。PTS/DTSは必ずしも全てのパケットヘッダに記録されている訳ではなく、MPEGに

よって記録するルールが規定されている。ルールの詳細についてはMPEGシステム (ISO/IEC 13818-1) 規格書に記述されているので省略する。

【0041】

パケットには更にヘッダ (パックヘッダ) が付けられ、パックを構成する。パックヘッダには、当該パックがいつデマルチプレクサを通過し、個々のエレメンタリストリームのデコーダバッファに入力されるかを示すタイムスタンプSCR (System Clock Reference) が記録されている。

(VOBのインターリーブ記録)

次に図11及び図12を用いてVOBファイルのインターリーブ記録について説明する。

【0042】

図11上段は、前述したプレーヤ構成図の一部である。図の通り、BDディスク上のデータは、光ピックアップを通してVOB即ちMPEGストリームであればトラックバッファへ入力され、PNG即ちイメージデータであればイメージメモリへと入力される。

トラックバッファはFIFOであり、入力されたVOBのデータは入力された順にデマルチプレクサへと送られる。この時、前述したSCRに従って個々のパックはトラックバッファから引き抜かれデマルチプレクサを介してビデオプロセッサまたはサウンドプロセッサへとデータが送り届けられる。一方で、イメージデータの場合は、どのイメージを描画するかはプレゼンテーションコントローラによって指示される。また、描画に使ったイメージデータは、字幕用イメージデータの場合は同時にイメージメモリから削除されるが、メニュー用のイメージデータの場合は、イメージメモリ内にそのまま残される。これはメニューの描画はユーザ操作に依存するところがあるため、同一イメージを複数回描画する可能性があるためである。

【0043】

図11下段は、BDディスク上でのVOBファイル及びPNGファイルのインターリーブ記録を示す図である。一般的にROM、例えばCD-ROMやDVD-ROMの場合、一連の連続再生単位となるAVデータは連続記録されている。これは、連続記録されている限り、ドライブは順次データを読み出しプレーヤ側に送り届けるだけで良いが、連続データが分断されてディスク上に離散配置されている場合は、個々の連続区間の間でシーク操作が入ることになり、この間データの読み出しが止まることになり、データの供給が止まる可能性があるからである。BD-ROMの場合も同様に、VOBファイルは連続領域に記録することができる方が望ましいが、例えば字幕データのようにVOBに記録されている映像データと同期して再生されるデータがあり、VOBファイルと同様に字幕データも何らかの方法によってBDディスクから読み出す必要がある。

【0044】

字幕データの読み出し方法の一手段として、VOBの再生開始前に一まとめで字幕用のイメージデータ (PNGファイル) を読み出してしまいう方法がある。しかしながら、この場合には一時記録に使用する大量のメモリが必要となり、非現実的である。

そこで、本実施の形態では、VOBファイルを幾つかのブロックに分けて、イメージデータとインターリーブ記録する方式を使用している。図11下段はそのインターリーブ記録を説明した図である。VOBファイルとイメージデータを適切にインターリーブ配置することで、前述したような大量の一時記録メモリ無しに、必要なタイミングでイメージデータをイメージメモリに格納することが可能になる。しかしながらイメージデータを読み出している際には、VOBデータの読み込みは当然のことながら停止することになる。

図12は、この問題を解決するトラックバッファを使ったVOBデータ連続供給モデルを説明する図である。

【0045】

既に説明したように、VOBのデータは、一旦トラックバッファに蓄積される。トラックバッファへのデータ入力レートとトラックバッファからのデータ出力レートの間に差を設けると、BDディスクからデータを読み出し続けている限り、トラックバッファのデータ蓄積量は増加をしていくことになる。ここでトラックバッファへの入力レートを V_a 、トラックバッファからの出力レートを V_b とする。図12の上段に記すようにVOBの一連続記録領域が論理アドレスの“a1”から“a2”まで続くとする。“a2”から“a3”の間は、イメージデータが記録されていて、VOBデータの読み出しが行えない区間であるとする。

図12の下段は、トラックバッファの内部を示す図である。横軸が時間、縦軸がトラックバッファ内部に蓄積されているデータ量を示している。時刻“t1”がVOBの一連続記録領域の開始点である“a1”の読み出しを開始した時刻を示している。この時刻以降、トラックバッファにはレート $V_a - V_b$ でデータが蓄積されていくことになる。このレートは言うまでもなくトラックバッファの入出力レートの差である。時刻“t2”は一連続記録領域の終了点である“a2”のデータを読み込む時刻である。即ち時刻“t1”から“t2”の間レート $V_a - V_b$ でトラックバッファ内はデータ量が増加していき、時刻“t2”でのデータ蓄積量は $B(t2)$ は下式によって求めることができる。

$$B(t2) = (V_a - V_b) \times (t2 - t1) \quad (\text{式1})$$

この後、BDディスク上のアドレス“a3”まではイメージデータが続くため、トラックバッファへの入力率は0となり、出力レートである“ $-V_b$ ”でトラックバッファ内のデータ量は減少していくことになる。これは読み出し位置“a3”まで、時刻でいう“t3”までになる。

ここで大事なことは、時刻“t3”より前にトラックバッファに蓄積されているデータ量が0になると、デコーダへ供給するVOBのデータが無くなってしまい、VOBの再生がストップしてしまう可能性がある。しかしながら、時刻“t3”でトラックバッファにデータが残っている場合には、VOBの再生がストップすることなく連続できることを意味している。

この条件は下式によって示すことができる。

$$B(t2) \geq -V_b \times (t3 - t2) \quad (\text{式2})$$

即ち、式2を満たすようにイメージデータの配置を決めればよい事になる。

(ナビゲーションデータ構造)

図13から図19を用いて、BD-ROMのナビゲーションデータ（BD管理情報）構造について説明をする。図13は、VOB管理情報情報ファイル（“YYY.VOBI”）の内部構造を示した図である。

【0046】

VOB管理情報は、当該VOBのストリーム属性情報（Attribute）とタイムマップ（TMAP）を有している。ストリーム属性は、ビデオ属性（Video）、オーディオ属性（Audio#0～Audio#m）個々に持つ構成となっている。特にオーディオストリームの場合は、VOBが複数本のオーディオストリームを同時に持つことができることから、オーディオストリーム数（Number）によって、データフィールドの有無を示している。

下記はビデオ属性（Video）の持つフィールドと夫々が持ち得る値である。

圧縮方式 (C o d i n g) :

M P E G 1
M P E G 2
M P E G 4

解像度 (R e s o l u t i o n) :

1 9 2 0 x 1 0 8 0
1 2 8 0 x 7 2 0
7 2 0 x 4 8 0
7 2 0 x 5 6 5

アスペクト比 (A s p e c t)

4 : 3
1 6 : 9

フレームレート (F r a m e r a t e)

6 0
5 9 . 9 4
5 0
3 0
2 9 . 9 7
2 5
2 4

下記はオーディオ属性 (A u d i o) の持つフィールドと夫々が持ち得る値である。

圧縮方式 (C o d i n g) :

A C 3
M P E G 1
M P E G 2
L P C M

チャンネル数 (C h) :

1 ~ 8

言語属性 (L a n g u a g e) :

タイムマップ (TMAP) はVOBU毎の情報を持つテーブルであって、当該VOBが有するVOBU数 (N u m b e r) と各VOBU情報 (VOBU# 1 ~ V O B U # n) を持つ。個々のVOBU情報は、VOBUの再生時間長 (D u r a t i o n) とVOBUのデータサイズ (S i z e) を夫々有している。

図14はVOBU情報の詳細を説明する図である。

【0047】

広く知られているように、MPEGストリームは時間的側面とデータサイズとしての側面との2つを有している。例えば、音声の圧縮規格であるAC3は固定ビットレートでの圧縮を行っているため、時間とアドレスとの関係は1次式によって求めることができる。しかしながらMPEGビデオデータの場合は、個々のフレームは固定の表示時間、例えばNTSCの場合は1フレームは1/29.97秒の表示時間を持つが、個々のフレームの

圧縮後のデータサイズは絵の特性や圧縮に使ったピクチャタイプ、いわゆる I/P/B ピクチャによってデータサイズは大きく変わってくる。従って、MPEG ビデオの場合は、時間とアドレスの関係は一般式の形で表現することは不可能である。

【0048】

当然の事として、MPEG ビデオデータを多重化している MPEG システムストリーム、即ち VOB も時間とデータとを一般式の形で表現することは不可能である。これに代わって、VOB 内での時間とアドレスとの関係を結びつけるのがタイムマップ (TMAP) である。図 14 に示すように、各 VOBU 毎に VOBU 内のフレーム数と、VOBU 内のパック数を夫々エントリーとして持つテーブルがタイムマップ (TMAP) である。

【0049】

図 15 を使って、タイムマップ (TMAP) の使い方を説明する。

図 15 に示すように時刻情報が与えられた場合、まずは当該時刻がどの VOBU に属するのかを検索する。これは、タイムマップの VOBU 毎のフレーム数を加算して行き、フレーム数の和が当該時刻を (フレーム数に換算して) 超えるまたは一致する VOBU が当該 VOBU になる。次にタイムマップの VOBU 毎のサイズを当該 VOBU の直前の VOBU まで加算して行き、その値が与えられた時刻を含むフレームを再生するために読み出すべきパックの先頭アドレスになっている。

【0050】

次に図 16 を使って、プレイリスト情報 ("XXX.PL") の内部構造を説明する。

プレイリスト情報は、セルリスト (Cell List) とイベントリスト (Event List) から構成されている。

セルリスト (Cell List) は、プレイリスト内の再生セルシーケンスであり、本リストの記述順でセルが再生される事になる。セルリスト (Cell List) の中身は、セルの数 (Number) と各セル情報 (Cell #1 ~ Cell #n) である。

【0051】

セル情報 (Cell #) は、VOB ファイル名 (VOB Name)、当該 VOB 内での有効区間開始時刻 (In) 及び有効区間終了時刻 (Out) と、字幕テーブル (Subtitle Table) を持っている。有効区間開始時刻 (In) 及び有効区間終了時刻 (Out) は、夫々当該 VOB 内でのフレーム番号で表現され、前述したタイムマップ (TMAP) を使うことによって再生に必要な VOB データのアドレスを得る事ができる。

【0052】

字幕テーブル (Subtitle Table) は、当該 VOB と同期再生される字幕情報を持つテーブルである。字幕は音声同様に複数の言語を持つことができ、字幕テーブル (Subtitle Table) 最初の情報も言語数 (Number) とそれに続く個々の言語ごとのテーブル (Language #1 ~ Language #k) から構成されている。

【0053】

各言語のテーブル (Language #) は、言語情報 (Language) と、個々に表示される字幕の字幕情報数 (Number) と、個々に表示される字幕の字幕情報 (Speech #1 ~ Speech #j) から構成され、字幕情報 (Speech #) は対応するイメージデータファイル名 (Name)、字幕表示開始時刻 (In) 及び字幕表示終了時刻 (Out) と、字幕の表示位置 (Position) から構成されている。

【0054】

イベントリスト (Event List) は、当該プレイリスト内であげられるイベントを定義したテーブルである。イベントリストは、イベント数 (Number) に続いて個々のイベント (Event #1 ~ Event #m) から構成され、個々のイベント (Event #) は、イベントの種類 (Type)、イベントの ID (ID)、イベント生成時刻 (Time) と有効期間 (Duration) から構成されている。

【0055】

図 17 は、個々のプレイリスト毎のイベントハンドラ (時間イベントと、メニュー選択

用のユーザイベント)を持つイベントハンドラテーブル("XXX. PROG")である。

イベントハンドラテーブルは、定義されているイベントハンドラ/プログラム数(Number)と個々のイベントハンドラ/プログラム(Program#1~Program#n)を有している。各イベントハンドラ/プログラム(Program#)内の記述は、イベントハンドラ開始の定義(<event_handler>タグ)と前述したイベントのIDと対になるイベントハンドラのID(ID)を持ち、その後に当該プログラムもFunctionに続く括弧"{"と"}"の間に記述する。

【0056】

次に図18を用いてBDディスク全体に関する情報("BD. INFO")の内部構造について説明をする。

BDディスク全体情報は、タイトルリスト(TitleList)とグローバルイベント用のイベントテーブル(EventTable)から構成されている。

タイトルリスト(TitleList)は、ディスク内のタイトル数(Number)と、これに続く各タイトル情報(Title#1~Title#n)から構成されている。個々のタイトル情報(Title)は、タイトルに含まれるプレイリストのテーブル(PLTable)とタイトル内のチャプタリスト(ChapterList)を含んでいる。プレイリストのテーブル(PLTable)はタイトル内のプレイリストの数(Number)と、プレイリスト名(Name)即ちプレイリストのファイル名を有している。

【0057】

チャプタリスト(ChapterList)は、当該タイトルに含まれるチャプタ数(Number)と個々のチャプタ情報(Chapter#1~Chapter#n)から構成され、チャプタ情報(Chapter#)は当該チャプタが含むセルのテーブル(CellTable)を持ち、セルのテーブル(CellTable)はセル数(Number)と個々のセルのエントリ情報(CellEntry#1~CellEntry#k)から構成されている。セルのエントリ情報(CellEntry#)は当該セルを含むプレイリスト名と、プレイリスト内でのセル番号によって記述されている。

【0058】

イベントリスト(EventList)は、グローバルイベントの数(Number)と個々のグローバルイベントの情報を持っている。ここで注意すべきは、最初に定義されるグローバルイベントは、ファーストイベント(FirstEvent)と呼ばれ、BDディスクがプレーヤに挿入された時、最初に呼ばれるイベントである。グローバルイベント用イベント情報はイベントタイプ(Type)とイベントのID(ID)だけを持っている。

【0059】

図19は、グローバルイベントハンドラのプログラムのテーブル("BD. PROG")である。本テーブルは、図17で説明したイベントハンドラテーブルと同一内容である。

(イベント発生メカニズム)

図20から図22を使ってイベント発生メカニズムについて説明する。

【0060】

図20はタイムイベントの例である。

前述したとおり、タイムイベントはプレイリスト情報("XXX. PL")のイベントリスト(EventList)で定義される。タイムイベントとして定義されているイベント、即ちイベントタイプ(Type)が"TimeEvent"の場合、イベント生成時刻("t1")になった時点で、ID"Ex1"を持つタイムイベントがシナリオプロセッサからプログラムプロセッサに対してあげられる。プログラムプロセッサは、イベントID"Ex1"を持つイベントハンドラを探し、対象のイベントハンドラを実行処理す

る。例えば、本実施例の場合では、2つのボタンイメージの描画を行うなどを行うことができる。

【0061】

図21はメニュー操作を行うユーザーイベントの例である。

前述したとおり、メニュー操作を行うユーザーイベントもプレイリスト情報("XXX.PL")のイベントリスト(Event List)で定義される。ユーザーイベントとして定義されるイベント、即ちイベントタイプ(Type)が"User Event"の場合、イベント生成時刻("t1")になった時点で、当該ユーザーイベントがレディとなる。この時、イベント自身は未だ生成されてはいない。当該イベントは、有効規格情報(Duration)で記される期間レディ状態にある。

【0062】

図21に描くように、ユーザがリモコンキーの「上」「下」「左」「右」キーまたは「決定」キーを押した場合、先ずUOPイベントがUOPマネージャによって生成されプログラムプロセッサに上げられる。プログラムプロセッサは、シナリオプロセッサに対してUOPイベントを流し、シナリオプロセッサはUOPイベントを受け取った時刻に有効なユーザーイベントが存在するかを検索し、対象となるユーザーイベントがあった場合は、ユーザーイベントを生成し、プログラムプロセッサに持ち上げる。プログラムプロセッサでは、イベントID"Ev1"を持つイベントハンドラを探し、対象のイベントハンドラを実行処理する。例えば、本実施例の場合では、プレイリスト#2の再生を開始する。

【0063】

生成されるユーザーイベントには、どのリモコンキーがユーザによって押されたかの情報は含まれていない。選択されたリモコンキーの情報は、UOPイベントによってプログラムプロセッサに伝えられ、仮想プレーヤが持つレジスタSPRM(8)に記録保持される。イベントハンドラのプログラムは、このレジスタの値を調べ分岐処理を実行することが可能である。

図22はグローバルイベントの例である。

【0064】

前述したとおり、グローバルイベントはBDディスク全体に関する情報("BD.INFO")のイベントリスト(Event List)で定義される。グローバルイベントとして定義されるイベント、即ちイベントタイプ(Type)が"Global Event"の場合、ユーザのリモコンキー操作があった場合にのみイベントが生成される。

ユーザが"メニュー"を押した場合、先ずUOPイベントがUOPマネージャによって生成されプログラムプロセッサに上げられる。プログラムプロセッサは、シナリオプロセッサに対してUOPイベントを流し、シナリオプロセッサは、該当するグローバルイベントを生成し、プログラムプロセッサに送る。プログラムプロセッサでは、イベントID"menu"を持つイベントハンドラを探し、対象のイベントハンドラを実行処理する。例えば、本実施例の場合ではプレイリスト#3の再生を開始している。

【0065】

本実施例では、単に"メニュー"キーと呼んでいるが、DVDのように複数のメニューキーがあってもよい。各メニューキーに対応するIDを夫々定義することで対応することが可能である。

(仮想プレーヤマシン)

図23を用いてプログラムプロセッサの機能構成を説明する。

【0066】

プログラムプロセッサは、内部に仮想プレーヤマシンを持つ処理モジュールである。仮想プレーヤマシンはBD-ROMとして定義された機能モデルであって、各BD-ROMプレーヤの実装には依存しないものである。即ち、どのBD-ROMプレーヤにおいても同様の機能を実行できることを保証している。

仮想プレーヤマシンは大きく2つの機能を持っている。プログラミング関数とプレーヤ変数（レジスタ）である。プログラミング関数は、Java（登録商標）Scriptをベースとして、以下に記す2つの機能をBD-ROM固有関数として定義している。

リンク関数：現在の再生を停止し、指定するプレイリスト、セル、時刻からの再生を開始する

```
Link (PL#, Cell#, time)
  PL#   : プレイリスト名
  Cell# : セル番号
  time  : セル内での再生開始時刻
```

PNG描画関数：指定PNGデータをイメージプレーンに描画する

```
Draw (File, X, Y)
  File : PNGファイル名
  X     : X座標位置
  Y     : Y座標位置
```

イメージプレーンクリア関数：イメージプレーンの指定領域をクリアする

```
Clear (X, Y, W, H)
  X : X座標位置
  Y : Y座標位置
  W : X方向幅
  H : Y方向幅
```

プレーヤ変数は、プレーヤの状態を示すシステムパラメータ（SPRM）と一般用途として使用可能なゼネラルパラメータ（GPRM）とがある。

図24はシステムパラメータ（SPRM）の一覧である。

SPRM (0)	: 言語コード
SPRM (1)	: 音声ストリーム番号
SPRM (2)	: 字幕ストリーム番号
SPRM (3)	: アンクル番号
SPRM (4)	: タイトル番号
SPRM (5)	: チャプタ番号
SPRM (6)	: プログラム番号
SPRM (7)	: セル番号
SPRM (8)	: 選択キー情報
SPRM (9)	: ナビゲーションタイマー
SPRM (10)	: 再生時刻情報
SPRM (11)	: カラオケ用ミキシングモード
SPRM (12)	: パレンタル用国情報
SPRM (13)	: パレンタルレベル
SPRM (14)	: プレーヤ設定値（ビデオ）
SPRM (15)	: プレーヤ設定値（オーディオ）
SPRM (16)	: 音声ストリーム用言語コード
SPRM (17)	: 音声ストリーム用言語コード（拡張）
SPRM (18)	: 字幕ストリーム用言語コード
SPRM (19)	: 字幕ストリーム用言語コード（拡張）

SPRM (20)	:	プレーヤーリジョンコード
SPRM (21)	:	予備
SPRM (22)	:	予備
SPRM (23)	:	再生状態
SPRM (24)	:	予備
SPRM (25)	:	予備
SPRM (26)	:	予備
SPRM (27)	:	予備
SPRM (28)	:	予備
SPRM (29)	:	予備
SPRM (30)	:	予備
SPRM (31)	:	予備

なお、本実施例では、仮想プレーヤーのプログラミング関数を Java (登録商標) Script ベースとしたが、Java (登録商標) Script ではなく、UNIX (登録商標) OS などで使われている B-Shell や、Perl Script など他のプログラミング関数であっても構わなく、言い換えれば、本発明は Java (登録商標) Script に限定されるものではない。

(プログラムの例)

図 25 及び図 26 は、イベントハンドラでのプログラムの例である。

【0067】

図 25 は、2つの選択ボタンを持ったメニューの例である。

セル (PlayList #1, Cell #1) 先頭でタイムイベントを使って図 25 左側のプログラムが実行される。ここでは、最初にゼネラルパラメータの一つ GPRM (0) に "1" がセットされている。GPRM (0) は、当該プログラムの中で、選択されているボタンを識別するのに使っている。最初の状態では、左側に配置するボタン 1 が選択されている事を初期値として持たされている。

【0068】

次に、PNG の描画を描画関数である Draw を使ってボタン 1、ボタン 2 夫々について行っている。ボタン 1 は、座標 (10, 200) を起点 (左端) として PNG イメージ "1black.png" を描画している。ボタン 2 は、座標 (330, 200) を起点 (左端) として PNG イメージ "2white.png" を描画している。

また、本セル最後ではタイムイベントを使って図 25 右側のプログラムが実行される。ここでは、Link 関数を使って当該セルの先頭から再度再生するように指定している。

【0069】

図 26 は、メニュー選択のユーザイベントのイベントハンドラの例である。

「左」キー、「右」キー、「決定」キー何れかのリモコンキーが押された場合夫々に対応するプログラムがイベントハンドラに書かれている。ユーザがリモコンキーを押した場合、図 21 で説明したとおり、ユーザイベントが生成され、図 26 のイベントハンドラが起動されることになる。本イベントハンドラでは、選択ボタンを識別している GPRM (0) の値と、選択されたりモコンキーを識別する SPRM (8) を使って分岐処理を行っている。

条件 1) ボタン 1 が選択されている、かつ、選択キーが「右」キーの場合
GPRM (0) を 2 に再設定して、選択状態にあるボタンを右ボタン 2 に変更する。

【0070】

ボタン 1、ボタン 2 のイメージを夫々書き換える。

条件 2) 選択キーが「決定 (OK)」の場合で、ボタン 1 が選択されている場合、プレ

イリスト#2の再生を開始する

条件3) 選択キーが「決定(OK)」の場合で、ボタン2が選択されている場合、プレイリスト#3の再生を開始する

上記のようにして実行処理が行われる。

(プレーヤ処理フロー)

次に図27から図30を用いてプレーヤでの処理フローを説明する。

【0071】

図27は、AV再生までの基本処理フローである。

BDディスクを挿入すると(S101)、BD-ROMプレーヤはBD.INFOファイルの読み込みと解析(S102)、BD.PROGの読み込み(S103)を実行する。BD.INFO及びBD.PROGは共に管理情報記録メモリに一旦格納され、シナリオプロセッサによって解析される。

【0072】

続いて、シナリオプロセッサは、BD.INFOファイル内のファーストイベント(FirstEvent)情報に従い、最初のイベントを生成する(S104)。生成されたファーストイベントは、プログラムプロセッサで受け取られ、当該イベントに対応するイベントハンドラを実行処理する(S105)。

ファーストイベントに対応するイベントハンドラには、最初に再生すべきプレイリスト情報が記録されていることが期待される。仮に、プレイリスト再生が指示されていない場合には、プレーヤは何も再生することなく、ユーザイベントを受け付けるのを待ち続けるだけになる。この場合、ユーザイベントを受け付けるのを待ち続けることになる(S201)。BD-ROMプレーヤはユーザからのリモコン操作を受け付けると、UOPマネージャはプログラムマネージャに対してUOPイベントを立ち上げる(S202)。

【0073】

プログラムマネージャは、UOPイベントがメニューキーによるものであるかを判別し(S203)、メニューキーの場合は、シナリオプロセッサにUOPイベントを流し、シナリオプロセッサがユーザイベントを生成する(S204)。プログラムプロセッサは生成されたユーザイベントに対応するイベントハンドラを実行処理する(S205)。

図28は、PL再生開始からVOB再生開始までの処理フローである。

【0074】

前述したように、ファーストイベントハンドラまたはグローバルイベントハンドラによってプレイリスト再生が開始される(S301)。シナリオプロセッサは、再生対象のプレイリスト再生に必要な情報として、プレイリスト情報"XXX.PL"の読み込みと解析(S302)、プレイリストに対応するプログラム情報"XXX.PROG"の読み込みを行う(S303)。続いてシナリオプロセッサは、プレイリストに登録されているセル情報に基づいてセルの再生を開始する(S304)。セル再生は、シナリオプロセッサからプレゼンテーションコントローラに対して要求が出される事を意味し、プレゼンテーションコントローラはAV再生を開始する(S305)。

【0075】

AV再生の開始(S401)を開始すると、プレゼンテーションコントローラは再生するセルに対応するVOBの情報ファイル(XXX.VOB.I)を読み込み及び解析をする(S402)。プレゼンテーションコントローラは、タイムマップを使って再生開始するVOBUとそのアドレスを特定し、ドライブコントローラに読み出しアドレスを指示し、ドライブコントローラは対象となるVOBデータを読み出し(S403)、VOBデータがデコーダに送られ再生が開始される(S404)。VOB再生は、当該VOBの再生区間が終了するまで続けられ(S405)、終了すると次のセル再生S304へ移行する。次にセルが無い場合は、再生が停止する(S406)。

【0076】

図29は、AV再生開始後からのイベント処理フローである。

BD-ROMプレーヤはイベントドリブン型のプレーヤモデルである。プレイリストの再生を開始すると、タイムイベント系、ユーザイベント系、字幕表示系のイベント処理プロセスが夫々起動され、平行してイベント処理を実行するようになる。

S500系の処理は、タイムイベント系の処理フローである。

【0077】

プレイリスト再生開始後(S501)、プレイリスト再生が終了しているかを確認するステップ(S502)を経て、シナリオプロセッサは、タイムイベント発生時刻になったかを確認する(S503)。タイムイベント発生時刻になっている場合には、シナリオプロセッサはタイムイベントを生成し(S504)、プログラムプロセッサがタイムイベントを受け取りイベントハンドラを実行処理する(S505)。

【0078】

ステップS503でタイムイベント発生時刻になっていない場合、または、ステップS504でイベントハンドラ実行処理後は再度ステップS502へ戻り、上述した処理を繰り返す。また、ステップS502でプレイリスト再生が終了したことが確認されると、タイムイベント系の処理は強制的に終了する。

S600系の処理は、ユーザイベント系の処理フローである。

【0079】

プレイリスト再生開始後(S601)、プレイリスト再生終了確認ステップ(S602)を経て、UOP受付確認ステップの処理に移る(S603)。UOPの受付があった場合、UOPマネージャはUOPイベントを生成し(S604)、UOPイベントを受け取ったプログラムプロセッサはUOPイベントがメニューコールであることを確認し(S605)、メニューコールであった場合は、プログラムプロセッサはシナリオプロセッサにイベントを生成させ(S607)、プログラムプロセッサはイベントハンドラを実行処理する(S608)。

【0080】

ステップS605でUOPイベントがメニューコールで無いと判断された場合、UOPイベントはカーソルキーまたは「決定」キーによるイベントである事を示している。この場合、現在時刻がユーザイベント有効期間内であるかをシナリオプロセッサが判断し(S606)、有効期間内である場合には、シナリオプロセッサがユーザイベントを生成し(S607)、プログラムプロセッサが対象のイベントハンドラを実行処理する(S608)。

【0081】

ステップS603でUOP受付が無い場合、ステップS606で現在時刻がユーザイベント有効期間に無い場合、または、ステップS608でイベントハンドラ実行処理後は再度ステップS602へ戻り、上述した処理を繰り返す。また、ステップS602でプレイリスト再生が終了したことが確認されると、ユーザイベント系の処理は強制的に終了する。

【0082】

図30は字幕処理のフローである。

プレイリスト再生開始後(S701)、プレイリスト再生終了確認ステップ(S702)を経て、字幕描画開始時刻確認ステップに移る(S703)。字幕描画開始時刻の場合、シナリオプロセッサはプレゼンテーションコントローラに字幕描画を指示し、プレゼンテーションコントローラはイメージプロセッサに字幕描画を指示する(S704)。ステップS703で字幕描画開始時刻で無いと判断された場合、字幕表示終了時刻であるかを確認する(S705)。字幕表示終了時刻であると判断された場合は、プレゼンテーションコントローラがイメージプロセッサに字幕消去指示を行い、描画されている字幕をイメージプレーンから消去する(S706)。

【0083】

字幕描画ステップS704終了後、字幕消去ステップS706終了後、または、字幕表示終了時刻確認ステップS705で当該時刻でないことが判断された場合、ステップS702に戻り、上述した処理を繰り返す。また、ステップS702でプレイリスト再生が終了したことが確認されると、字幕表示系の処理は強制的に終了する。

(実施例2)

次に本発明の第2の実施の形態について説明する。

【0084】

第2の実施の形態は、BDアプリケーションにおいてより豊かなインタラクティブ性を実現するため、Java（登録商標）のようなプログラミング環境を導入することに関する内容である。基本的には第1の実施例に基づく内容であり、拡張または異なる部分を中心に説明する。

(プログラミング環境の導入)

実施例1において、図4が示す「BD再生プログラミング」の主な目的は、AVデータの再生順序を規定する、あるいは、ユーザーの操作によりAVの再生順序を変更することであった。しかし、BD再生プログラミングとしてJava（登録商標）のような汎用的なプログラミング環境を利用するのであれば、AV再生の制御のみならず様々なアプリケーションを提供することが可能である。本実施例では、汎用的なプログラミング環境としてJava（登録商標）を想定しているが、C言語やその他のプログラミング言語であっても、同様である。

【0085】

AV再生以外のアプリケーションが利用可能になったとしても、あるカテゴリーごとにAV再生やアプリケーションを認識する方が、ユーザーには分かりやすくなる。カテゴリーとは映画本編や本編以外の付属のコンテンツとして提供される各種ゲームといったカテゴリーであり、カテゴリーをユーザーに認識できるようにするため、タイトルという単位を用意することとする。

【0086】

実施例1においては、図18が示すとおり、タイトルとは再生順序が記述されているプレイリストへのリンク情報であり、タイトルが指定されると関連付けられたプレイリストを参照して、プレイリストが参照しているセルやAVストリームが再生された。

本実施例においてタイトルが指定されると、プログラムを介したAVストリームが再生される場合もあれば、AVストリームを再生しないアプリケーションが起動する場合もある。

【0087】

図31は図6を本実施例に合わせて拡張したものである。ディスクより読み込まれたプログラム(3101)は、プログラム処理部(3102)で処理される。プログラムは管理情報処理部やプレゼンテーション処理部に指示を出し、プレイリストを介してAVストリームを再生するものもあれば、通常のJava（登録商標）アプリケーション同様にイメージプレーン(3104)に対して描画指示(3105)を行うものもある。

【0088】

AVストリームの再生のみを指示するアプリケーションの例としては、本編映画再生をユーザーの指示に従って行う単純な再生アプリケーションが挙げられる。AV再生を行わないアプリケーションの例としては、フル画面をプログラムで描画するゲームのようなアプリケーションが挙げられる。AVを再生しながらアプリケーションとして描画することも可能であり、再生中のAVの上にプログラムが描画したイメージ図を重ねたり、移動させてアニメーションを行ったりすることも可能である。

【0089】

タイトルはユーザーに認識されるAVストリームの再生及びアプリケーションの実行の単位であるが、同時にプレーヤがAVストリームの再生及びアプリケーションの実行を管

理する単位でもある。

(アプリケーションの管理)

図 3 2 はタイトルリスト (3 2 0 1) が、アプリケーションの管理単位としてのタイトルを指し示している様子を示している。

【0 0 9 0】

タイトル # 1 は映画本編が再生されるタイトルである。タイトル # 1 が選択されると、本編映像再生用のアプリケーション (3 2 0 2) が起動され、本編映像の再生が開始される。本編再生用のアプリケーションは、リモコンキーによる早送り・スキップなどのイベントを受けて、再生制御部に指示を出しプレーヤの動作をコントロールする。タイトル # 1 が起動されると本編再生用アプリケーションとともに、ポップアップメニューを出現させるためのアプリケーション (3 2 0 3) や、本編再生中の映像に出てきた品物を購入できるようにショッピングカートアプリケーション (3 2 0 4) などとも起動させる。

【0 0 9 1】

本編再生から別の目的のタイトルに移りたい場合は、タイトル切替を行う。たとえば本編再生中に出てきた品物を購入するためにオンラインショッピングアプリケーションを起動させたい場合は、オンラインショッピングを行うタイトル # 2 に移動する。

タイトル # 1 からタイトル # 2 に移動すると、タイトル # 1 の場合同様にタイトル # 2 に関連したアプリケーション (3 2 0 5) が起動されるとともに、タイトル # 2 に関係ないアプリケーション (3 2 0 2、3 2 0 3) は停止する。タイトル # 1 にもタイトル # 2 にも関係するアプリケーション (3 2 0 4) は、どちらのタイトルでも起動し続けることができ、タイトル # 1 とタイトル # 2 の間のタイトル遷移中に停止と再起動が起こるわけではない。

【0 0 9 2】

タイトル # 3 はタイトル # 1 及びタイトル # 2 とは独立したタイトルであり、タイトル # 3 が選択されると、タイトル # 1 及びタイトル # 2 で起動されていたアプリは停止し、タイトル # 3 に関連したアプリケーション (3 2 0 6) が起動される。

図 3 3 はこれまで述べたような、タイトル単位でのアプリケーションの管理を実現するための管理情報について示している。

【0 0 9 3】

アプリケーション管理情報 (3 3 0 1) は、タイトル番号 (3 3 0 2) とアプリケーション ID (3 3 0 3) のペアを管理しており、指定されたタイトル中でどのアプリケーションが起動されるかを管理している。図 3 3 中の例では、タイトル # 1 において、アプリケーション # 1 とアプリケーション # 2 が起動され、タイトル # 2 においてアプリケーション # 3 が起動される。

【0 0 9 4】

タイトルが選択されるとすぐにアプリケーション管理情報にしたがって関連するアプリケーションが起動され (3 3 0 4)、タイトルが変わると次のタイトルで関連付けられていないアプリケーションは終了し (3 3 0 5)、次のタイトルに関連付けられているアプリケーションが起動される (3 3 0 6)。

なお、アプリケーション ID としては、Applet や xlet などの起動インターフェースを持ったクラスファイル名などでよい。

図 3 4 は、拡張することによりタイトル開始時の起動状態を設定可能にしたアプリケーション管理情報について示している。

【0 0 9 5】

図 3 3 で示されたアプリケーション管理情報は、あるタイトルが開始されたときには必ず関連付けられたアプリケーションが起動された。そこでタイトルが開始されたときに起動されるアプリケーションと起動されないアプリケーションを区別するために、タイトル開始時の起動状態 (3 4 0 1) を持たせることにする。

アプリケーション#1はタイトル#1に開始時に自動的に起動されるアプリケーションである。アプリケーション#2は同じくタイトル#1に関連付けられているが、タイトル開始時に自動的に起動はされない。アプリケーション#2は、他のアプリケーションによって起動される(3402)、あるいはユーザー操作などのイベントにより起動される。

【0096】

アプリケーション#2は、タイトル#1開始時に自動的に起動されることはないが、タイトル#2に切り替わるときに起動していて、かつ、タイトル#2に登録されていなければ、自動的に終了する。

アプリケーションの終了の仕方としては、アプリケーション#2のように他のアプリやイベントによって起動する場合と同様、他のアプリやイベントによって終了することもある(3404)。このような場合、起動属性がAutoRun(自動的に起動)であっても、タイトル中で自動的に再起動することはない。タイトルが切り替わり、再びタイトル#1が開始されたときは、自動的に起動される。

【0097】

図35は、アプリケーション管理情報の起動状態の拡張を行い、複数タイトル間で継続的にアプリケーションを起動しておく方法、アプリケーションのサスペンド状態を実現する方法を説明するためのものである。

複数のタイトルに対して、同じアプリケーションを登録しておくことにより、タイトルを越えてアプリケーションを起動させ続けることが可能となる。アプリケーション管理情報において、アプリケーション#1はタイトル#1にもタイトル#2にも登録(3501、3502)されている。アプリケーション#1はタイトル#1開始時にAutoRun属性により自動的に起動される。タイトル#1からタイトル#2に遷移する際、アプリケーション#1はタイトル#2にも登録されているため、タイトル#1終端で終了することなく、タイトル#2に遷移する。このように複数タイトルに同一アプリケーションを登録しておけば、登録されているタイトル間を遷移する際にアプリケーションが終了後再起動されるようなことはない。

【0098】

複数タイトルをまたげるアプリケーションであったとしても、登録されていないタイトルに遷移する際は終了される。アプリケーション#1の例では、タイトル#2からタイトル#3に遷移する際は、アプリケーション#1は終了される。次に、アプリケーションをサスペンド状態に遷移させる方法を説明する。通常の起動状態では、Java(登録商標)VMなどの実行環境において、実行環境内のメモリを割り当てられ、CPUリソースも割り当てられることにより実行されている。起動していないアプリケーションは、メモリもCPUリソースも割り当てられていない。それに対して、サスペンド状態とは、実行環境内のメモリに格納されているが、CPUリソースは割り当てられていない状態であり、CPUリソースを割り当てられれば、いつでも実行可能となる。このようなサスペンド状態を管理するために、起動属性として「サスペンド」を追加する。起動属性としてサスペンドを指定されているアプリケーションは、起動していればサスペンド状態に遷移する。

【0099】

図35においてアプリケーション#2がサスペンドされるアプリケーションである。アプリケーション#2はタイトル#1において起動され、タイトル#2に遷移する際に自動的にサスペンド状態となる。アプリケーション#2はタイトル#2中において、実行環境のメモリ内には存在できるがCPUリソースは割り当てられない状態である。タイトル#3のアプリケーション管理情報では、アプリケーション#2は通常状態(サスペンドではない)の起動属性を持っているため、タイトル#2からタイトル#3に遷移する際にアプリケーション#2はレジュームされ、再び起動状態に戻る。

【0100】

図36は、起動属性によって、タイトル間遷移時に状態がどのように変化するかを示した図表である。これまでの例として述べてきたように、アプリケーションが起動していな

い状態で起動属性としてAutoRunが指定されているタイトルに遷移すると、アプリケーションは自動的に起動される。すでにアプリケーションが起動された状態で起動属性がサスペンドであるタイトルに遷移すると、アプリケーションはサスペンドされる。アプリケーションがサスペンドされた状態で、起動属性がAutoRunあるいは指定されていない（サスペンドではない）タイトルに遷移すると、アプリケーションはレジュームされ、起動状態に戻る。それ以外の場合はアプリケーションの状態に変化は起こらず、前の状態をそのまま継続する。

【0101】

なお、アプリケーションは他のアプリケーションからの制御やイベントを受けてアプリケーション自体が反応することによって、アプリケーションの状態を変更することもある。図34のアプリケーション#2は、他のアプリケーションからの起動される例である。

なお、アプリケーションがサスペンドされている状態で、起動属性が指定されていないタイトルに遷移した際、レジュームするのではなく何もしなくてもよい。図37はアプリケーション管理情報の管理単位の区分を示している。各单位ごとの管理情報は、単位ごと（タイトルの場合はタイトルごと、図37の例）に管理情報を分割して別々に記録（3701）してもよいし、タイトルごとの管理情報でもまとめて図35のアプリケーション管理情報ように管理してもよい。

【0102】

ここまでのアプリケーション管理情報は、カテゴリの単位としてのタイトルを基本とし、タイトル単位でアプリケーションの起動や終了を管理していた。アプリケーションによっては複数のタイトルをまたげるものもあるが、ディスクに存在する全てのタイトルで有効なアプリケーションの場合は、全ての管理テーブルに書かなければならず、煩雑になる。そこで、ディスク全体で有効なアプリケーションに関しては、ディスク単位の管理情報（3702）を用意すれば、ディスクで一括して管理できるため管理が容易になる。

【0103】

同様に、ディスク3枚組のようなパッケージがある場合は、ディスクセット単位でアプリケーションを管理するアプリケーション管理情報（3703）を、どのディスクにも同じ情報を記録しておくことで実現可能である。

また、これらの階層別のアプリケーション管理情報を組み合わせて使うことも可能である。組み合わせて利用する場合で、どちらか一方にしか登録されていない場合（たとえばディスクのアプリケーション管理情報には登録されていて、タイトルのアプリケーション管理情報には登録されていない場合）は、管理テーブルの和集合をとってプレーヤ内で独自のアプリケーション管理情報を構成すればよい。2つの管理テーブルで情報が異なる場合は、より有効範囲が狭い管理テーブル（先ほどの例では、タイトルのアプリケーション管理情報）を優先するようにしてもよい。

【0104】

複数のディスクにまたがっても、アプリケーションを一意に指定するため、アプリケーションIDとして、オーサーが決めた独自の番号だけではなく、オーサーごとに割り振られた一意なIDとオーサーが決めた独自の番号を組み合わせることにより、他社製のコンテンツとアプリケーションIDが重なることを防ぐことも可能である。

なお、複数ディスクにまたがる管理情報の場合、ディスクが抜かれた状態でのアプリケーションの動作を規定することも可能である。ディスクが抜かれた状態ではアプリケーションを一時停止状態にする、あるいは、使用できる機能を限定しなければ、悪意を持ったアプリケーションやバグにより、ディスク交換時あるいはディスクを抜いたあともアプリケーションが起動し続け、ユーザーが気づかないうちにデータの改変あるいはネットワークへのアクセスを行うかもしれない。

【0105】

図38はタイトルより短い単位でアプリケーションを管理する方法を示している。AVストリームが再生されるタイトルの場合、タイトル再生中にいくつかのチャプター（3801）が用意されていることがある。チャプターはAVストリームの再生時間軸上に順番

に並んでいるため、時間軸上の特定の区間でのみ有効なアプリケーションを規定したい場合、タイトル指定とともにさらに時間の限定を加えてチャプターあるいはチャプターの範囲を指定する(3802)ことにより、アプリケーションの有効期間をタイトル中の時間軸上の特定期間に限定する(3803)ことができる。

【0106】

なお、さらに細かい単位でアプリケーションを管理したい場合には、タイトル内のストリームの時間軸を利用し、再生開始時点よりA秒後からB秒後のように時間指定することも可能である。

(アプリケーションマネージャ)

アプリケーション管理情報を利用して、アプリケーションを起動及び終了させる仕組みについて述べる。

【0107】

図39はプログラミング環境を用いて再生を制御する部分の大まかなブロック図である。ディスク(3901)から読み込まれたプログラムやプログラムから使用するイメージデータは、ローカルメモリ(3902)に蓄積される。ローカルメモリは図6のプログラム記憶メモリと同等であるが、プログラムだけではなくイメージデータなども格納される。ローカルメモリ上のプログラム及びデータは、Java(登録商標)仮想マシン(3903)などのプログラム処理部が必要とした時に、プログラム処理部のワークメモリ(3904)内に読み込まれる。ワークメモリはプログラムを実行する際にプログラム処理部が利用するメモリであり、Java(登録商標)仮想マシンではヒープ領域あるいはヒープメモリなどと呼ばれるメモリである。ワークメモリに読み込まれプログラム処理部によりプログラムが実行されるとアプリケーション(3905)が起動される。図6におけるプログラム処理部は、Java(登録商標)仮想マシンとワークメモリが含まれている。アプリケーションの動作によっては、プログラム処理部を介して、AVストリーム再生部(3906)に指示を出すことも可能である。

【0108】

アプリケーション管理情報は、アプリケーションマネージャ(3907)によって読み込まれ、アプリケーションマネージャがプログラム処理部を制御することにより、アプリケーションの起動や終了を行う。タイトル遷移時に、アプリケーションマネージャはアプリケーション管理情報を参照し、該当するアプリケーションのプログラムをローカルメモリより読み込んで実行、あるいは必要ないアプリケーションを終了する。

【0109】

なお、ディスクからローカルメモリにデータが読み込まれる時、あるいは、ローカルメモリからワークメモリにデータが読み込まれるときに、プログラムやイメージデータのサイズをチェック、プログラムのコードにデータ落ちなどが存在しないか、プログラムを動作させるJava(登録商標)仮想マシンとバージョンが合っているかなどチェックする機能があってもよい。また、不正なプログラムが動作することを防ぐためプログラムを暗号化してディスクに記録しておき、読み込む際に複合化するなどセキュリティを確保する機能があってもよい。

図40はアプリケーションマネージャによるアプリケーションの起動及び終了の様子を示している。

【0110】

タイトル管理情報を参照してタイトル開始時にアプリケーションマネージャがアプリケーションの起動する(4001)以外に、ユーザー操作により発生するイベントや時間イベントに応じて、アプリケーションマネージャがアプリケーションを起動(4002)してもよい。また、他のアプリケーションからの制御によりアプリケーションを起動する(4003)ことも可能である。この場合、アプリケーションが直接他のアプリを起動してもよいし、アプリケーションマネージャに指定のアプリケーションを起動するように問い

合わせてもよい。

【0111】

アプリケーションの終了は、タイトル開始時と同様、タイトル終了時にアプリケーションマネージャが次のタイトルで登録されていないアプリケーションIDを持つアプリを終了(4004)させる、あるいは、ユーザー操作により発生するイベントや時間イベントに応じてアプリケーションを終了(4005)させてもよい。他のアプリケーションから終了(4006)してもよい。この場合、直接アプリケーションを終了させてもよいし、アプリケーションマネージャに、指定のアプリケーションを終了するように問い合わせてもよい。

【0112】

図41はアプリケーションマネージャによるアプリケーションの終了方法について示している。アプリケーションを終了させる際は、アプリケーションマネージャより終了させたいアプリケーションに対して終了イベントを通知し、アプリケーションのイベント受信部(4101)がイベントを受け、アプリケーション内の終了プロセス(4102)を実行することによりアプリケーションが自ら終了する。アプリケーションが終了イベント受信部と対応した終了プロセスを持たない(4103)ため自ら終了しない、あるいは、終了プロセスが間違っておりアプリケーションが自ら終了しない場合などは、一定時間経過後アプリケーションマネージャがアプリケーションを強制的に終了させる。

【0113】

この方法はサスペンド時やレジューム時にも応用可能であり、アプリケーションマネージャがアプリケーションを強制的にサスペンドあるいはレジュームする前に、アプリケーション独自のプロセスを処理する仕組みを提供することが可能である。

また、アプリケーションマネージャは、アプリケーション自体に終了やサスペンドの処理を任せるのだけではなく、アプリケーションが終了やサスペンドなどのイベントに正常に反応しなかった場合、強制的に処理することにより、個々のアプリケーションだけではなく全体の動作を管理することが可能となる。この仕組みにより、たとえあるタイトル内でアプリケーションの動作がおかしくなったとしても、タイトルを切り替えることによりアプリケーションマネージャにより登録されていないアプリケーションは終了され、アプリケーションの管理を保証する仕組みを提供することが可能となる。全てのアプリケーションを終了させるために、どのアプリケーションも実行されないタイトルを用意しておくことにより、そのタイトルに遷移することで全てのアプリケーションを停止することも可能である。このようなアプリケーションも動作できない、あるいは安全な特別なアプリケーションしか起動できないタイトルは、アプリケーションの動作がおかしくなったときに正常状態に復帰する手段として有効である。映画コンテンツなどの場合は、メニューや再生開始時の注意書きを表示するタイトルなどを利用するとよい。

(メモリ管理)

図42はプログラムやプログラムが使用するイメージデータなどが蓄積されるローカルメモリと、プログラムが実行されるときにデータを読み込んでおくワークメモリの関係と役割について示している。

【0114】

ローカルメモリはディスク上のデータを一時的に蓄えておくキャッシュとしての役割を果たしている。ディスクにはAVストリームとプログラムやイメージデータが記録されているが、AVストリームを再生時に起動するアプリケーションのプログラムを読み込んでいては、ディスク上でシークが発生してしまい、最悪AVストリームの再生が一時的に停止してしまうかもしれない。そこで、AVストリームの再生を開始する前や再生が一時的に停止してもよい点において、アプリケーションを起動するために必要なプログラムやデータを読み込み蓄積しておくことにより、AVストリームの再生中はAV再生のため以外のシークが発生させないようにする。

【0115】

なお、ハードディスクドライブのようにヘッドのシークやディスクの回転待ちが短時間である場合は、必要になったときにプログラムやデータを読み込み、必要最低限のサイズを持ったローカルメモリにデータを蓄積しておいてもよい。

アプリケーションマネージャがアプリケーションを起動しようとする際、ローカルメモリより必要なプログラムをワークメモリに読み込んで J a v a (登録商標) 仮想マシン内で実行する。アプリケーション起動中は、アプリケーションが必要とするプログラム及びデータは、基本的にローカルメモリにすでに読み込まれていることが望ましい。J a v a (登録商標) 仮想マシンから直接ディスク上のファイルを読み込むことはなく、プログラムやデータは必ずローカルメモリ上に蓄積される。

【0116】

アプリケーションが終了すると、J a v a (登録商標) 仮想マシンはワークメモリから不要なデータを解放 (ガーベッジコレクション) する。ただし、アプリケーションが終了したとしてもローカルメモリから自動的にデータが削除されるわけではない。アプリケーションが再び起動するときにはローカルメモリから読み込む必要があるため、アプリケーションが必要なくなるまで、ローカルメモリにデータを保持しておく、ディスクから再び同じデータを読み込む必要がなくなり効率がよい。

(仮想ドライブ)

ローカルメモリは J a v a (登録商標) 仮想マシンから仮想ドライブのようにアクセス可能なメモリである。ディスクより読み込まれたデータはローカルメモリ内でディレクトリ構造などを構成し、アプリケーションからは R A M メモリドライブ上のファイルのようにアクセス可能である。

【0117】

図 4 3 はディスク上のファイルとローカルメモリ上のファイルを管理する仕組みを示している。(A) のような管理情報があれば、ディスク上のファイルをローカルメモリ上に読み込んだ際、ローカルメモリ上ではどのディレクトリのなんというファイル名で扱えばよいか、規定することが可能である。また (B) のように、読み込むべきファイルの一覧作り、ローカルメモリ内でのファイル名は、データとともにヘッダ情報内で管理してもよい。このような仕組みでディスク上のデータは、ローカルメモリ上に配置される。ただし、ファイル数が多くなると、管理が煩雑になってしまう。

【0118】

なお、ファイル名の代わりに新たにロケータを定義してもよい。ローカルメモリだけではなく、ハードディスクなどの高速なアクセスが可能なストレージとローカルメモリを統一的に扱えるロケータを定義することにより、少ないメモリで多くのデータを読み込んでおくことが可能である。ハードディスクなどのストレージとローカルメモリの管理をプレーヤの実装にまかせるならば、メモリのページングなどの技術も導入可能である。

【0119】

そこで図 4 4 のように、ローカルメモリ上でのファイル名やディレクトリ構造も含めて、複数のファイルをひとまとめにして管理することができる構造を利用することで、ディスク上からは 1 つのファイルを読み込むだけで、ローカルメモリ上で複数のファイルやディレクトリを構成可能とする。このようにディレクトリ構造を含めて複数のファイルをひとまとめに扱えるデータ構造として、t a r ファイルや z i p ファイルを利用してもよい。

【0120】

なお、ファイルの解放の容易性を考慮し、ディレクトリごとにデータをまとめ、データを解放するときは、ディレクトリごと解放する方にしてもよい。

(データ読み込みのタイミング)

ワークメモリのデータの読み込み及び解放は、アプリケーション管理情報にしたがって行われる。アプリケーションの起動及び終了を指示するトリガーとなるのはアプリケーション

ョンマネージャであり、実際にデータを読み込んで実行するのは J a v a (登録商標) 仮想マシンである。

【0121】

それに対し、データがディスクからローカルメモリに読み込まれるタイミングは、データが必要になる期間に依存する。データが必要となる期間は、アプリケーションがデータを必要とする期間であり、タイトルなどのアプリケーションの管理単位と同じ単位で管理される。

図45はデータの読み込み及びローカルメモリからの解放のタイミングを決める、データ管理情報(4501)を示した図である。データ管理情報は、タイトル番号(4502)と該当するタイトルで必要となるデータ名あるいはデータを固めて保持しているデータブロック名(4503)のペアが管理されている。あるタイトルが選択されると、選択されたタイトルで必要となるデータをローカルメモリに読み込み(4504)、タイトル切替によって別のタイトルに遷移する際に、遷移先のタイトルに登録されていないデータはローカルメモリより解放(4505)する。

【0122】

なお、読み込み及び解放のタイミングは、アプリケーション管理情報と同様に、タイトル単位のほか、ディスク単位やディスクセット単位、あるいはタイトルより小さな単位として、チャプターや時間指定も可能である。

また、有効期間ごとにデータをまとめて、一つのファイルとしてディスクに配置しておく、データ管理情報が簡潔になる。

(リソースの管理)

アプリケーション管理情報を利用することにより、タイトル中で実行されるアプリケーション数やアプリケーションの実行に必要なワークメモリの容量を有限なサイズに抑えることが可能となる。アプリケーション管理情報がなければ、プログラムの勘違いやプログラムのバグ・誤動作により、アプリケーションが想定数以上に起動してしまい、ワークメモリに入りきらず、全てのアプリケーションの動作が不安定になるかもしれない。ワークメモリなどのリソースを制限するため、プログラムが間違っていたとしても指定されたアプリケーション以外は起動しないように、アプリケーション管理情報及びアプリケーションマネージャによる制御が有効である。

【0123】

アプリケーションの起動をアプリケーション自身に任せている場合、プログラムを解析してどのアプリケーションがどの時点で起動しているか判断しなければならず、また、全ての条件を把握しなければ正確に状態をつかめない、デバッグ及びベリファイすることが非常に困難である。それに対し、アプリケーション管理情報はプログラム内ではなく静的なデータとして保存されているため、どのタイトル中にどのアプリケーションが起動されるあるいは起動される可能性があるか容易に把握でき、各アプリケーションごとに必要となるリソースを見積もることができれば、タイトル中に必要なリソースのトータルサイズも見積もることが可能となる。

【0124】

データ管理情報も同様であり、タイトル中にローカルメモリに読み込まれるデータサイズを容易に割り出すことができるため、ローカルメモリのサイズ内に収まるかどうか判断することができ、ベリファイに有効に利用することが可能である。

この管理方法は、ディスクで配布されるコンテンツやアプリケーションが利用するメモリなどのリソースのサイズを常に一定とする方法である。図46のように、年月が経つにつれてメモリの容量が大容量化され、プレーヤに搭載されたメモリ容量が大きくなったとしても、ディスクは古いプレーヤで搭載されていたメモリ容量で全てのアプリケーションが動作するように作らなければ、そのディスクが全てのプレーヤで動作することが保証されない。あるいは、大容量のメモリをめいっぱい利用するアプリケーションを記録したディスクがあったとしても、メモリを少ししか搭載されていないプレーヤではそのディス

クに記録されたアプリケーションを実行できなくなってしまう。図中のデータ#3及びデータ#4は古いプレーヤではローカルメモリに読み込むことができず、イメージを表示したりプログラムを実行することができない。そのため、使用できるメモリ容量の最大値を制限することは、全てのプレーヤでの動作保証には有効だが、プログラム実行環境の進化においていかれることになりかねない。

【0125】

アプリケーション管理情報やデータ管理情報を拡張、リソースの最低必須サイズを規定することにより、メモリが少ないプレーヤでも動作保証され、メモリが多いプレーヤでは余分にデータを読み込むことにより、データ読み込みのレスポンスを良くしてユーザーの操作を快適にしたり、メモリが少ないプレーヤでは実行できないあるいは品質の悪かったアプリケーションが、同じディスクを使ってもメモリの多いプレーヤでは実行できるとあるいは品質が良くなる仕組みを導入する。

【0126】

図46を用いて説明すると、データ#1やデータ#2は映画を再生する、あるいは必ず実行されなければならないアプリケーションのデータであり、これらのデータは古いプレーヤでも新しいプレーヤでも必ずローカルメモリ上に読み込まれる。それに対し、データ#3やデータ#4のデータは、大容量のメモリが積まれている新しいプレーヤでのみ実行されればよい、おまけ的なプログラムである。

【0127】

このような仕組みにより、どのプレーヤでも最低限のコンテンツの再生やアプリケーションの実行を保証でき、メモリサイズの大容量化などリソースの進化に対応できる仕組みを提供することが可能である。

なお、ワークメモリの管理はローカルメモリのように行うのは困難である。ローカルメモリはディスクから読み込まれたデータがそのまま配置されるため、データサイズを容易に見積もることが可能であるが、ワークメモリはJava（登録商標）仮想マシンのヒープ領域であり、読み込まれたプログラムのデータ以外にプログラムを管理するデータや変数や配列のために確保したメモリも考慮しなければならず、Java（登録商標）仮想マシンの実装による影響が大きい。そのため、ワークメモリに関してはメモリのサイズによって動作保証を行うのではなく、アプリケーションが規定サイズのイメージデータを表示可能なことや、規定個数のスレッドを生成できるといった最低限必要な機能を満たすことで保証するのが望ましい。あるいは、ワークメモリの上限値をオーバーしてアプリケーションがメモリを利用しようとしたときに、オーバーフローしていることを通知するイベントあるいは例外処理を発生させ、ワークメモリが不足したときの対策をアプリケーションに行わせる方法も可能である。

（優先度情報を用いたデータ読み込み）

図47はデータ管理情報の拡張を行い、ローカルメモリサイズに応じて読み込むデータを選択する仕組みについて示している。データ管理情報に有効期間としてのタイトル番号、そのタイトル中で読み込まれるデータに対応させて、データの読み込み優先度情報（4701）を追加する。もっとも単純な優先度情報は、図表中の例のように、必ず読み込まなければいけないデータ（mandatory）とメモリに余裕があれば読み込む（optional）の2つを区別できる情報であればよい。

【0128】

タイトル#1において、データ#1とデータ#2はローカルメモリに必ず読み込まなければならないデータ（4702）である。これらのデータとして、映画の再生を行うアプリケーションはメニューを表示するアプリケーションなど、どのプレーヤでも実行されなければならないようなプログラムなどが該当する。読み込み必須のデータの合計サイズは、どのプレーヤでも必ず搭載されていることが保証されている最小ローカルメモリサイズ以内に納める必要がある。プレーヤの最小ローカルメモリサイズが規定されていれば、アプリケーション制作者はプログラムのバイトコードのサイズ及びアプリケーションで利用

するイメージデータなどの合計サイズが、最小ローカルメモリサイズに収まるように調整することができ、サイズが決まれば作成後にベリファイすることも可能である。

【0129】

データ#3はローカルメモリに余裕があれば読み込まれるデータ(4703)であり、これらのデータとしては、おまけのゲームや直接本編の再生やシナリオの分岐に関わらないアプリケーションであり、実行されなくてもメインコンテンツの再生に影響がないアプリケーションなどである。読み込みオプションのデータは、ローカルメモリに読み込み必須のデータを読み込んだ後にまだ空き容量がある場合、空き容量とデータサイズを比較し空き容量の方が大きければ、ディスクからローカルメモリに読み込まれる。メモリが少ない場合は読み込まれない。図48はデータ読み込みの優先度情報をさらに細かくしたものを示している。複数の読み込みオプションのデータがある場合、複数のデータを全て読み込めるだけのローカルメモリが搭載されていれば問題はない。複数存在するデータのうち一部のデータを読み込むだけのメモリ空き容量がなければ、どの読み込みオプションデータを読み込むか判断する必要がある。この順序は、プレーヤごとに違う動作をしてもよいが、データ管理テーブルにデータが出現する順序で読み込んでよい。後者の場合、同じローカルメモリが搭載されていれば、同じデータが読み込まれることになる。

【0130】

あるいは、読み込みオプションのデータに対してはより細かい優先度(4801)を設け、メモリに空き容量がある場合、優先度が高いものからメモリに読み込めるか判定を行う。より優先度が高いデータを読み込める場合はそのデータを読み込み、さらに空き容量がある場合に限り次の優先度を持つデータを読み込めるか判定を行う。

図49はオプションデータの読み込み判定のフローを示している。タイトルが遷移してデータ管理情報による読み込みが始まる(S4901)と、まず読み込み必須のデータをローカルメモリに読み込む(S4902)。読み込み必須データ以外に、読み込み判定を行っていない読み込みオプションデータがあるかチェックを行い(S4903)、読み込み判定を行っていないデータがあれば、判定を行っていないデータのうちの最も優先度が高いデータに対して、ローカルメモリの空き容量とデータのサイズの比較を行う(S4904)。比較の結果、データサイズの方が大きければローカルメモリに読み込めないため、次の優先度のデータの判定に移行する。空き容量の方が大きければ、データをローカルメモリに読み込み(S4905)、次の優先度のデータの判定に移行する。全ての読み込みオプションデータの読み込みチェックが終了すれば、判定を終了する(S4906)。

(アプリケーション管理情報とデータ管理情報の統合)

アプリケーション管理情報はアプリケーションの有効期間を管理しており、ワークメモリ上のデータの生存期間を管理している。データ管理情報はデータの読み込み及び解放のタイミングを管理しており、ローカルメモリ上でのデータの生存期間を管理している。どちらも独立して管理可能であるが、アプリケーション及びデータの管理単位はタイトルなど同じ単位を基本としているため、生存期間が同じものは情報を結合させ、一元的に管理することも可能である。また、アプリケーションを実行するためにワークメモリにデータを読み込むためには、ローカルメモリにデータがなければならないので、アプリケーションに必要なデータは同じ生存期間で管理した方が簡潔である。

【0131】

図50はアプリケーション管理情報にデータ管理情報を統合した、アプリケーション・データ管理情報を示している。アプリケーションは実行するために必要なデータと関連付けられるため、タイトル及びアプリケーションIDと読み込みデータを関連付けて管理情報を統合している。

なお管理情報が統合されたため、データの読み込み及び解放のタイミングや、ローカルメモリの制御、ディスクからローカルメモリへのデータの読み込み指示もアプリケーションマネージャが行ってもよいし、別のモジュールが管理してもよい。

【0132】

アプリケーション起動及び終了、データの読み込み・解放の基本的な管理単位はタイトルである。図中の例で説明する。タイトル#1に切り替わりタイトルが開始されると、読み込み必須のデータをローカルメモリに読み込み、空き容量がある場合は読み込める分だけ読み込みオプションデータを読み込む。データを読み込み終わるとアプリケーションの起動属性にしたがい、アプリケーションマネージャがアプリケーションを起動させる。アプリケーション#1はデータ読み込み優先度が読み込み必須であり、必ずローカルメモリにデータが読み込まれている。また、起動属性がAutoRunであるため、タイトル開始時に自動的にアプリケーションが起動される。アプリケーション#2は読み込みオプションであるため、メモリに余裕がある場合はローカルメモリに読み込まれているが、メモリが少ない場合はローカルメモリに読み込まれていない。

【0133】

図51のアプリケーション・データ管理情報は、起動属性と読み込み優先度に相関関係を作り、より単純な静的データといくつかのルールにより読み込みの制御を実現する、別の形式の管理情報である。ルールの例としては、起動属性がAutoRunのアプリケーションは、必ず指定の期間中起動されるとルールを決め、それらアプリケーションのデータは必ずローカルメモリに読み込まれることとする。また、AutoRunではないアプリケーションに関しては、有効期間が大きいものほど優先度が高い（図中の例ではアプリケーション#3の方がアプリケーション#4より優先度が高い）とし、優先順位にしたがってローカルメモリの空き容量に合わせて、該当するアプリケーションのデータを読み込む。

【0134】

なお、これらの優先順位は、起動属性に関係なく、アプリケーションの有効期間が大きいものほど、あるいは、アプリケーションの開始期間が早いものほど優先されるとしてもよいし、単に起動属性だけで優先順位付けしてもよい。また、アプリケーション・データ管理情報の登録順をデータの優先順位としてもよいし、それらの組み合わせでもよい。

特に映画コンテンツのような場合、AVストリームを再生するためのアプリケーションは必ず実行されなければならないため、それら必ず起動されなければならないアプリケーションに対しては、起動属性としてさらに特別な属性（5101）を持たし、起動を保証するようにしてもよい。チャプターのようなAVストリームの時間軸上に定義されるような区間を利用する場合は、時間軸を決めるAVストリームを再生するアプリケーションが起動しなければ、ほかのアプリケーションを制御できなくなってしまうかもしれない。メモリの状態によりほかのアプリが起動されなくても、必ず起動されなければならないアプリケーションに、AutoRun (mandatory) のような特別な起動属性を持たせると再生が保証される。

（アプリケーション起動時の例外処理）

全てのデータが必ず読み込まれているモデルではなく、ローカルメモリの容量に応じて読み込まれないデータが存在する場合、アプリケーションマネージャや他のアプリケーションからの制御により、ローカルメモリにデータがないアプリケーションの起動指示があり得る。

【0135】

図52はデータの読み込み、アプリケーション起動の指示の様子を順を追って示している。アプリケーション・データ管理情報にしたがって、ローカルメモリにデータを読み込んでいく。このときデータ#3（5201）は優先度が低く、ローカルメモリの容量が少なかったため読み込まれなかった。にもかかわらず、タイトル中でユーザーの操作などによりアプリケーション#3の起動が促されると、Java（登録商標）仮想マシンはワークメモリにデータを読み込むためにローカルメモリにアクセスする。このときのアプリケーションの読み込み命令に対し、ローカルメモリにデータがないことを確認したアプリケーションマネージャあるいはローカルメモリを管理するモジュールは、必要とされるデータの優先度を最大にし、使われておらず、読み込み必須ではないデータをローカルメモリ

中から削除してでも、必要なデータをディスクから読み出しにいくことも可能である。

アプリケーションがほかのアプリケーションの起動を指示した場合、ローカルディスクに起動しようとするアプリケーションのデータがなければ、アプリケーションの起動命令に対して次の3つのような動作が起こりえる。

1) アプリケーションは、ディスクからローカルメモリにデータが読み込まれ、ローカルメモリからワークメモリへのデータの読み込みが完了するまで待つ。ローカルメモリへのデータアクセスがアプリケーションから行われた結果、ディスクからローカルメモリへのデータの読み込みはアプリケーションの制御ではなく、アプリケーションマネージャあるいはローカルメモリを管理するモジュールが行う。

【0136】

2) 起動命令（読み込み命令）にタイムアウトを設け、起動するまでに一定時間待機、その後それでもデータがローカルメモリから読み込めない場合は、読み込みに失敗と判断して、読み込み失敗時の例外処理を行う。

3) ローカルメモリにデータがない時点ですぐに読み込み失敗と判断、読み込み失敗時の例外処理を行う。

前述の3つのどの動作をするかは、アプリケーションのプログラム中にある、ほかのプログラム起動命令と合わせてプログラムしておけばよい。映画コンテンツの場合、AVストリームの再生中に1)あるいは2)のタイプのデータの読み込みが発生すると、データを読み込むためにディスク上でシークが発生するかもしれない。その場合、AVストリームの供給が一時的に止まるため、AVストリームの再生が一時的に停止してしまうかもしれない。

【0137】

アプリケーションマネージャによるアプリケーション起動指示の場合、プログラムによる任意の例外処理はできないが、上記のどの方法をとることも可能である。ただし、アプリケーションマネージャが起動指示を出す性質上、アプリケーションのデータがローカルメモリに読み込まれていないのであれば、ローカルメモリの容量が足りなかった場合であり、ほかのデータを削除して起動するのはアプリケーション・データ管理情報の優先度情報と矛盾する動作である。そのため、2)あるいは3)の動作が望ましく、AVストリームの再生中ならばAVストリームの再生を優先し、シークが発生させないことが望ましい。

【0138】

ローカルメモリが少ない場合は、アプリケーション起動のためデータ読み込みが発生しAVストリームの再生が一時的に停止する場合がある。ローカルメモリが多いとシークが発生せず、連続したAVストリームの再生が行える、あるいはレスポンスを早くできる利点がある。

ローカルメモリのサイズを固定せず拡張性を持たせる仕組みをアプリケーション・データ管理情報の優先度情報などの仕組みを提供することで、必要最低限のローカルメモリしか搭載していないプレーヤでも必ず再生あるいは実行されなければならないアプリケーションの動作保証を行い、大容量のローカルメモリを搭載したプレーヤはユーザーに快適なアプリケーションの動作を提供できる仕組みが実現可能となる。

（AVストリームと同時にデータを提供する仕組み）

ここまで述べてきたデータ提供のタイミングは、タイトル開始時のAVストリームの再生が始まる前にデータをプリロードしておく方法、あるいは、アプリケーションがデータを実用としたときにAVストリームの再生を一時停止してでもデータを読みに行く方法であった。十分大容量のローカルメモリを搭載しておけば、タイトル選択中に必要となるデータを全てプリロードしておくことは可能であるが、必ずしも大容量を搭載しているわけではない。また、データを読み込むためにはディスク上でシークが発生してしまうため、

AVストリームの再生が一時的に停止してしまう。そのため、図53のような場合、アプリケーション・データ管理情報上は十分なローカルメモリを備えているにもかかわらず、有効に活用できなくなってしまう。

【0139】

データ#1(5301)とデータ#2(5302)は同時にローカルメモリに読み込むことが可能であり、データ#1とデータ#3(5303)も同時にローカルメモリに読み込むことが可能である。ただし、データ#1とデータ#2とデータ#3を同時に読み込むだけの容量をローカルメモリは持っておらず、全てのデータをAVストリームの再生開始時(タイトル開始時)にプリロードすることはできない。そのため、チャプター#2が終了してからチャプター#4が開始されるまでの間に、データ#2をローカルメモリから解放し、データ#3をディスクから読み出してくる必要がある。AVストリームと別ファイルで配置してあるとシークが発生してしまい、AVストリームの再生が一時停止してしまうかもしれないが、AVストリームが映画コンテンツなどの場合、再生が一時的に停止することは望ましくない。そこで、AVストリームを再生中にシークをすることなく、データをローカルメモリに読み込む方法として、多重化方式あるいはインターリーブ方式を用いる。

【0140】

図54は多重化方式で、データをAVストリームと一緒に多重化する方法である。多重化すべきJava(登録商標)プログラムやイメージデータ(5401)は、データを管理する情報とともにセクション構造(5402)に内包する。セクションをMPPEGストリーム多重化するために、セクション構造をPESパケット(5403)に内包する。PESパケット(Packetized Elementary Streamパケット)は188バイト単位のTSパケット(Transport Streamパケット、5404)に分割され、同様にTSパケットに分割されたビデオ(5405)及びオーディオ(5406)ストリームとともに多重化されて、1本のストリーム(5407)を構成する。

【0141】

このように多重化されたデータは、ストリームをTSパケットに分解し、PESパケットに含まれるセクション、セクションに含まれるデータを抽出することで取り出すことが可能であり、また、AVストリームと同時のデータを読み込むことが可能である。

図55はデータをインターリーブ配置することにより、AVデータと交互にデータを読み込む方法を示している。データ(5501)はデータの管理情報とともにインターリーブユニット(5502)に内包する。インターリーブユニットに内包するデータは1つでもよいし複数でもよい。インターリーブユニットはAVストリームファイル(5503)の途中で割り込ませて配置(インターリーブ配置)され、AVストリームの開始点やAVストリームとインターリーブユニットとの境界点のアドレス情報を保持しておく。境界点情報から、どのアドレスのデータが、AVストリームかデータかを判定することが可能である。データを読み込む際は、AVストリームとインターリーブユニットを交互に続けて読み込み、読み込み中のアドレスがAVストリームの場合はAVストリームを処理する部分にデータを供給し、読み込み中のアドレスがインターリーブユニットを保持するアドレスであれば、インターリーブユニットを読み込み、内包されているデータを取り出して、ローカルメモリに格納する。

【0142】

このような方法により、AVストリームの再生中であっても、データをローカルメモリに読み込むことが可能となり、図53の例であってもチャプター#3再生中に、シークが発生させることなく連続してデータを読み込み続けるだけで、データ#3をローカルメモリに読み込むことが可能となる。

インターリーブ方式ではAVストリームとデータ(インターリーブユニット)の境界ははっきりしているため、シークによりデータだけ読み出すことは容易であるが、多重化方式では、ストリーム中にデータが分散して配置されてしまっているため、データだけ読み

込むためであったとしても、ストリーム全部を解析しデータ部分を取り出す必要があり、データの読み込みが非常に複雑な作業となる。そこで、AVストリーム再生中はストリームとともに送られてくるデータを利用し、ローカルメモリに余裕がある場合にはAVストリームとは別のファイルとして配置されている同じデータからデータを読み出す仕組みを導入する。

【0143】

図56は、AVストリーム再生中にシークを発生させることなくデータを読み込むための多重化方式あるいはインターリーブ方式のデータと、ローカルメモリが大容量なためプリロードするためにAVストリームとは別ファイルで配置してある同じデータを管理するための、アプリケーション・データ管理情報の例を示している。ストリーム中に多重化あるいはインターリーブ配置されているデータの管理情報(5601)と別ファイルでは位置してあるデータの管理情報(5602)が2つ登録されているが、読み込まれるデータは同一である。

【0144】

ストリーム中のデータは、タイトル開始時あるいはAVストリーム開始時にディスクからローカルメモリにデータをプリロードすることはできない。優先度が読み込み必須であったとしても、プリロードするのではなく、データが必要になる前に、AVストリーム再生していると自動的にローカルメモリに読み込まれる。この例ではアプリケーション#3で必要になるデータはチャプター#4以前に読み込まれていなければならないので、ストリーム中に埋め込まれたデータはチャプター#4以前に配置しなければならない。データの配置は、一度だけではなく、複数回同じデータを埋め込むことにより、AVストリームの再生を一部スキップしたとしても、再び出現したデータを読み込むことにより、データを読み込み損ねることを回避することも可能である。

【0145】

ローカルメモリが大容量な場合は、低い優先度のデータまでプリロードすることができるため、アプリケーション#1及びアプリケーション#2及びアプリケーション#3のデータをすべてローカルメモリに読み込むことが可能となる。その場合は最初からアプリケーション#3で必要なデータも読み込まれているため、ストリーム抽出する必要はなく、また、AVストリームのジャンプにより、直接チャプター#4に移動してもアプリケーションを起動することが可能となる。

(サイズが違う同じデータ名のデータの複数配置)

図57は同じアプリケーションで必要なデータをローカルメモリのサイズに合わせて、イメージデータの解像度や品質を向上させる仕組みを、アプリケーション・データ管理情報で実現する方法を示している。

【0146】

アプリケーション#1はプログラムとともにイメージデータを必要とするアプリケーションである。タイトル開始時にアプリケーション#1用の読み込み必須データ(5701)をまずローカルメモリに読み込む。読み込み必須データは使用するイメージデータの品質を落とし、サイズを小さくしてある。ローカルメモリにまだ空き容量がある場合は、同じアプリケーション#1用のデータで、次の優先度を持つもの(5702)を読み込もうとする。こちらのデータはイメージデータとしてそれなりの品質を持っているため、先ほどのイメージデータよりサイズが大きくなっている。ただし、これらのデータはディスク上のデータ名は異なっているが、ローカルメモリ上に展開される際は、同じファイル名になるように工夫してある。そのため、データが読み込まれることにより、品質の悪いイメージデータは品質がそれなりのイメージデータで上書きされる。同様に、より優先度は低いかれどイメージデータの品質がよくデータサイズが大きなデータ(5703)をローカルメモリに読み込むことが可能であれば、さらにイメージの品質を上げることが可能となる。このような管理方法により、ローカルメモリの容量が大きければ大きいほど、品質の高いアプリケーションを実行できる仕組みを提供可能となる。

【0147】

なお、同じファイル名のデータを複数回配置する場合、後から読み込まれたデータを優先して以前のデータを上書きしてもよいし、先に読み込まれたデータを優先して一度データを読むと削除されるまで同じファイル名のデータは読み込まないようにしてもよい。

(排他関係の明示)

アプリケーションによっては、排他関係にあるアプリケーションが存在するかもしれない。図58の例のように、アプリケーション#2とアプリケーション#3はともにアプリケーション#1より起動されるが、両方が同時に起動されることはない。このようなアプリケーションとして、アプリケーション#1がアプリケーション起動ランチャー、アプリケーション#2がゲーム、アプリケーション#3が学習用アプリケーションのようなものがあげられる。アプリケーション起動ランチャーからはどちらか1つしか起動できず、どちらかのアプリケーションが起動中はランチャーが操作できないため、他方を起動することができない。

【0148】

このような場合、ローカルメモリ(5801)内は、アプリケーション#1のデータとアプリケーション#2のデータ、あるいは、アプリケーション#1のデータとアプリケーション#3のデータが共存することはあっても、3つのアプリケーションが同時にローカルメモリに読み込まれることはない。これらの状態を管理するために、アプリケーション・データ管理情報に互いにはいた関係にあるアプリケーションを識別するための情報を加えることにより、ローカルメモリのベリファイが容易になる。

(管理情報の配置)

アプリケーション・データ管理情報はディスク上のルートディレクトリに配置しておいてもよいし、タイトルごとに管理情報を分け、タイトルごとのディレクトリに配置してもよい。AVストリームごとに管理情報を分け、AVストリームの先頭に配置、あるいは、ストリームのヘッダ情報として埋め込んでおいてもよい。

【0149】

なお、AVストリームのヘッダ情報としてアプリケーション・データ管理情報を配置する場合、優先度の判定はストリームが読み込まれるまで分からないため、タイトル開始時に優先度判定を行うのではなく、ストリームを読み込んでヘッダ情報からアプリケーション・データ管理情報を検出するたびに行ってもよい。(AVストリームの再生の保証)

タイトルが選択されたときに、どのようにアプリケーションが起動されるか述べてきたが、アプリケーションがマスターとなってAVを再生するモデルであった。そのため、AV再生のマスターとなるアプリケーションが正常に動作するために、どのように動作保証するかが最大の問題となった。

【0150】

図59はAVストリームの再生だけは、Java(登録商標)仮想マシン上のアプリケーションで管理しない例を示している。タイトルリスト(5901)よりタイトルが選択されると、アプリケーションマネージャ(5902)がアプリケーション・データ管理情報(5903)を参照し、アプリケーションの起動及び終了を行う。アプリケーションマネージャによりJava(登録商標)仮想マシン(5904)上に起動されたアプリケーションが再生制御部(5905)に指示を出すことにより、AVストリームを再生することも可能である。タイトルリストの各タイトルはそのタイトル中で再生されるPlayList(5906)の情報を保持しており、アプリケーションが起動しなくてもJava(登録商標)仮想マシン上のアプリケーションとは別の再生制御モジュール(5907)によって、PlayListから参照されるAVストリームの再生を開始することも可能である。

【0151】

このような仕組みにより、たとえばアプリケーションが起動しなかったりバグなどにより

暴走したとしても、A Vストリームの再生を担うモジュールは独立して動作し、A Vストリームの再生を保証することが可能となる。

なお、この場合は、再生制御モジュールが再生開始したA Vストリームに対しても、アプリケーションから制御できることが望ましい。あるいは、アプリケーションが再生を開始したA Vストリームであっても、何らかの異常でアプリケーションが停止してしまった場合、自動的に再生制御モジュールがA V再生の制御を継承し、A Vストリームの再生を継続することも可能である。

【産業上の利用可能性】

【0152】

本発明によれば、映像データとプログラムとを含むコンテンツの開発にプログラミング環境を導入するため、特に、映像コンテンツの制作に携わる映画産業・民生機器産業において利用される可能性をもつ。

【図面の簡単な説明】

【0153】

- 【図1】DVDの構成図
- 【図2】ハイライトの構成図
- 【図3】DVDでの多重化の例を示す図
- 【図4】BD-ROMのデータ階層図
- 【図5】BD-ROM上の論理空間の構成図
- 【図6】BD-ROMプレーヤの概要ブロック図
- 【図7】BD-ROMプレーヤの構成ブロック図
- 【図8】BD-ROMのアプリケーション空間の説明図
- 【図9】MPEGストリーム（VOB）の構成図
- 【図10】バックの構成図
- 【図11】A Vストリームとプレーヤ構成の関係を説明する図
- 【図12】トラックバッファへのA Vデータ連続供給モデル図
- 【図13】VOB情報ファイル構成図
- 【図14】タイムマップの説明図
- 【図15】タイムマップを使ったアドレス情報取得方法説明図
- 【図16】プレイリストファイルの構成図
- 【図17】プレイリストに対応するプログラムファイルの構成図
- 【図18】BDディスク全体管理情報ファイルの構成図
- 【図19】グローバルイベントハンドラを記録するファイルの構成図
- 【図20】タイムイベントの例を説明する図
- 【図21】ユーザイベントの例を説明する図
- 【図22】グローバルイベントハンドラの例を説明する図
- 【図23】仮想マシンの構成図
- 【図24】プレーヤ変数テーブルの図
- 【図25】イベントハンドラ（タイムイベント）の例を示す図
- 【図26】イベントハンドラ（ユーザイベント）の例を示す図
- 【図27】プレーヤの基本処理のフローチャート
- 【図28】プレイリスト再生処理のフローチャート
- 【図29】イベント処理のフローチャート
- 【図30】字幕処理のフローチャート
- 【図31】プログラム実行環境を鑑みたプレーヤの概要ブロック図
- 【図32】アプリケーション管理単位の説明図
- 【図33】アプリケーション管理情報の説明図
- 【図34】起動状態情報を備えたアプリケーション管理情報の説明図
- 【図35】起動状態情報に休止状態を備えたアプリケーション管理情報の説明図
- 【図36】起動属性によるタイトル間状態遷移の説明図

- 【図 37】 階層的なアプリケーション管理単位の説明図
- 【図 38】 チャプター単位の管理単位を備えたアプリケーション管理情報の説明図
- 【図 39】 データ及びアプリケーションの管理に関わるプレーヤの概要ブロック図
- 【図 40】 アプリケーションの起動及び終了の説明図
- 【図 41】 イベントを利用したアプリケーション終了方法の説明図
- 【図 42】 データの読み込み及び解放タイミングの説明図
- 【図 43】 ディスク上とメモリ上のデータを関連付けるデータ管理情報を示す図
- 【図 44】 複数ファイルを結合して管理するデータ構造の説明図
- 【図 45】 アプリケーション管理とデータの読み込みのタイミングの関連の説明図
- 【図 46】 メモリサイズの拡張性と動作保証についての説明図
- 【図 47】 データ読み込み優先度情報を備えたアプリケーション管理情報を示す図
- 【図 48】 細かい区分の優先度情報を備えたアプリケーション管理情報を示す図
- 【図 49】 優先度情報を用いたデータ読み込み判定のフローチャート
- 【図 50】 アプリケーションとデータの管理を統合した管理情報を示す図
- 【図 51】 データ読み込み優先度情報をアプリケーション起動属性情報に従属させた管理情報を示す図
- 【図 52】 アプリケーション起動の流れの説明図
- 【図 53】 プリロードのみではメモリを有効に使えない例を示す図
- 【図 54】 データをAVストリームとともに多重化した構造図
- 【図 55】 データをAVストリームにインタリーブ配置した構造図
- 【図 56】 ストリーム中のデータと別ファイルのデータを扱った管理情報を示す図
- 【図 57】 優先度情報利用しメモリサイズに応じてデータの品質を向上する説明図
- 【図 58】 排他的に起動されるアプリケーションの識別情報を備えた管理情報を示す図

- 【図 59】 プログラム実行部とAVストリーム再生部を分けた概要ブロック図
- 【符号の説明】

【0154】

- | | |
|-----|-----------------|
| 201 | BDディスク |
| 202 | 光ピックアップ |
| 203 | プログラム記録メモリ |
| 204 | 管理情報記録メモリ |
| 205 | AV記録メモリ |
| 206 | プログラム処理部 |
| 207 | 管理情報処理部 |
| 208 | プレゼンテーション処理部 |
| 209 | イメージプレーン |
| 210 | ビデオプレーン |
| 211 | 合成処理部 |
| | |
| 301 | プログラム記録メモリ |
| 302 | プログラムプロセッサ |
| 303 | UOPマネージャ |
| 304 | 管理情報記録メモリ |
| 305 | シナリオプロセッサ |
| 306 | プレゼンテーションコントローラ |
| 307 | クロック |
| 308 | イメージメモリ |
| 309 | トラックバッファ |
| 310 | デマルチプレクサ |
| 311 | イメージプロセッサ |

- 312 ビデオプロセッサ
- 313 サウンドプロセッサ
- 314 イメージプレーン
- 315 ビデオプレーン
- 316 合成処理部
- 317 ドライブコントローラ

- 3101 プログラム
- 3102 プログラム処理部
- 3103 プログラム処理部から管理情報処理部の制御
- 3104 イメージプレーン
- 3105 プログラム処理部からイメージプレーンの制御

- 3201 タイトルリスト
- 3202 本編映像データを再生するアプリケーション
- 3203 ポップアップメニューを表示するアプリケーション
- 3204 オンラインショッピング購入品を蓄えておくアプリケーション
- 3205 オンラインショッピングを実現するアプリケーション
- 3206 ゲームアプリケーション

- 3301 アプリケーション管理情報
- 3302 タイトル番号
- 3303 アプリケーションID
- 3304 アプリケーション#1の起動
- 3305 アプリケーション#1の終了
- 3306 アプリケーション#3の起動

- 3401 起動属性
- 3402 アプリケーション#1によるアプリケーション#2の起動
- 3403 タイトル終端におけるアプリケーションの終了
- 3404 アプリケーション#2によるアプリケーション#1の終了

- 3501 タイトル#1におけるアプリケーション#1の登録情報
- 3502 タイトル#2におけるアプリケーション#1の登録情報

- 3701 タイトル単位のアプリケーション管理情報
- 3702 ディスク単位のアプリケーション管理情報
- 3703 ディスクセット単位のアプリケーション管理情報

- 3801 チャプター
- 3802 チャプター単位で管理されたアプリケーション管理情報
- 3803 チャプター単位で管理されたアプリケーション

- 3901 ディスク
- 3902 ローカルメモリ
- 3903 Java (登録商標) 仮想マシン
- 3904 ワークメモリ
- 3905 Java (登録商標) アプリケーション
- 3906 再生制御部
- 3907 アプリケーションマネージャ

- 4001 アプリケーションマネージャによるタイトル開始時の起動
- 4002 アプリケーションマネージャによるイベント発生時の起動
- 4003 アプリケーションによる起動
- 4004 アプリケーションマネージャによるタイトル終了時の終了
- 4005 アプリケーションマネージャによるイベント発生時の終了
- 4006 アプリケーションによる終了

- 4101 終了イベント受信部
- 4102 終了プロセス
- 4103 終了イベント及び終了プロセスを持たないアプリケーション

- 4501 データ管理情報
- 4502 タイトル番号
- 4503 読み込みデータ名
- 4504 ディスクからローカルメモリへのデータの読み込み
- 4505 ローカルメモリからデータの解放

- 4701 読み込み優先度情報
- 4702 必ずローカルメモリに読み込まれるデータ
- 4703 必ずしもローカルメモリには読み込まれないデータ

- 4801 細かい区分の読み込み優先度情報

- 5101 起動属性に統合された読み込み必須属性

- 5201 読み込み優先度の低いデータ

- 5301 タイトル全体で有効なデータ
- 5302 チャプター#1から#2の間で有効なデータ
- 5303 チャプター#4から#5の間で有効なデータ

- 5401 データ
- 5402 セクション
- 5403 Packetized Elementary Stream
パケット
- 5404 Transport Stream パケット
- 5405 ビデオデータ
- 5406 オーディオデータ
- 5407 多重化されたストリーム

- 5501 データ
- 5502 インターリーブユニット
- 5503 AVストリーム
- 5601 ストリーム中に多重化あるいはインターリーブ配置
されたデータ
- 5602 ストリームとは別ファイルとして配置されたデータ

- 5701 優先度が高くサイズが小さいデータ
- 5702 優先度が並でサイズが並のデータ

5703 優先度が低くサイズが大きいデータ

5801 ローカルメモリ

5901 タイトルリスト

5902 アプリケーションマネージャ

5903 アプリケーション・データ管理情報

5904 Java (登録商標) 仮想マシン

5905 再生制御部

5906 プレイリスト

5907 再生制御モジュール

S101 ディスク挿入ステップ

S102 BD. INFO読み込みステップ

S103 BD. PROG読み込みステップ

S104 ファーストイベント生成ステップ

S105 イベントハンドラ実行ステップ

S201 UOP受付ステップ

S202 UOPイベント生成ステップ

S203 メニューコール判定ステップ

S204 イベント生成ステップ

S205 イベントハンドラ実行ステップ

S301 プレイリスト再生開始ステップ

S302 プレイリスト情報 (XXX. PL) 読み込みステップ

S303 プレイリストプログラム (XXX. PROG)
読み込みステップ

S304 セル再生開始ステップ

S305 AV再生開始ステップ

S401 AV再生開始ステップ

S402 VOB情報 (YYY. VOB I) 読み込みステップ

S403 VOB (YYY. VOB) 読み込みステップ

S404 VOB再生開始ステップ

S405 VOB再生終了ステップ

S406 次セル存在判定ステップ

S501 プレイリスト再生開始ステップ

S502 プレイリスト再生終了判定ステップ

S503 タイムイベント時刻判定ステップ

S504 イベント生成ステップ

S505 イベントハンドラ実行ステップ

S601 プレイリスト再生開始ステップ

S602 プレイリスト再生終了判定ステップ

S603 UOP受付判定ステップ

S604 UOPイベント生成ステップ

S605 メニューコール判定ステップ

S606 ユーザーイベント有効期間判定ステップ

- S 6 0 7 イベント生成ステップ
- S 6 0 8 イベントハンドラ実行ステップ

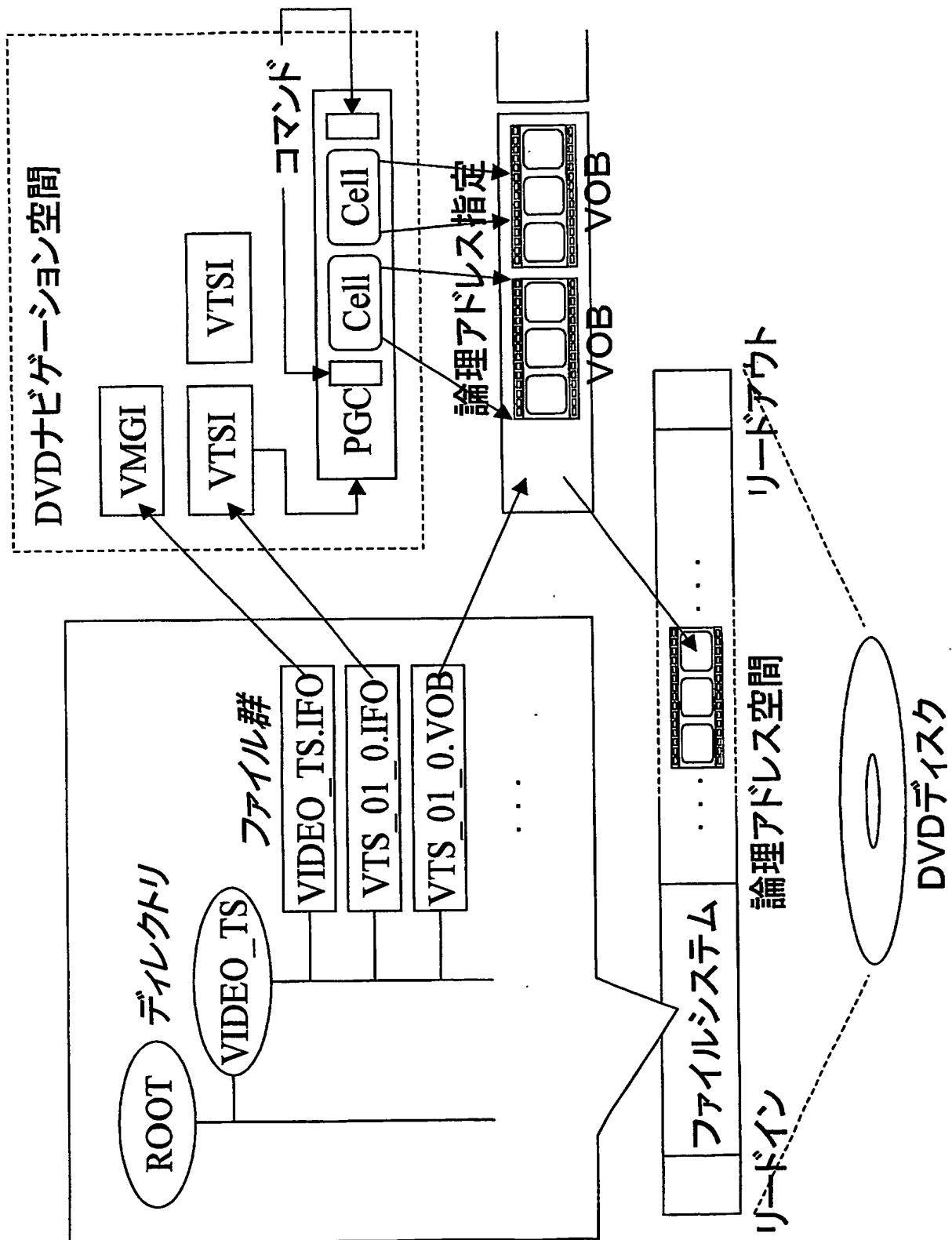
- S 7 0 1 プレイリスト再生開始ステップ
- S 7 0 2 プレイリスト再生終了判定ステップ
- S 7 0 3 字幕描画開始判定ステップ
- S 7 0 4 字幕描画ステップ
- S 7 0 5 字幕表示終了判定ステップ
- S 7 0 6 字幕消去ステップ

- S 1 0 0 1 ディスク挿入ステップ
- S 1 0 0 2 BD. CLASS 検出判定ステップ
- S 1 0 0 3 BDオブジェクト生成ステップ
- S 1 0 0 4 イベントハンドラ宣言ステップ
- S 1 0 0 5 BD. INFO 読み取りステップ
- S 1 0 0 6 ファーストイベント生成ステップ
- S 1 0 0 7 イベントハンドラ処理ステップ

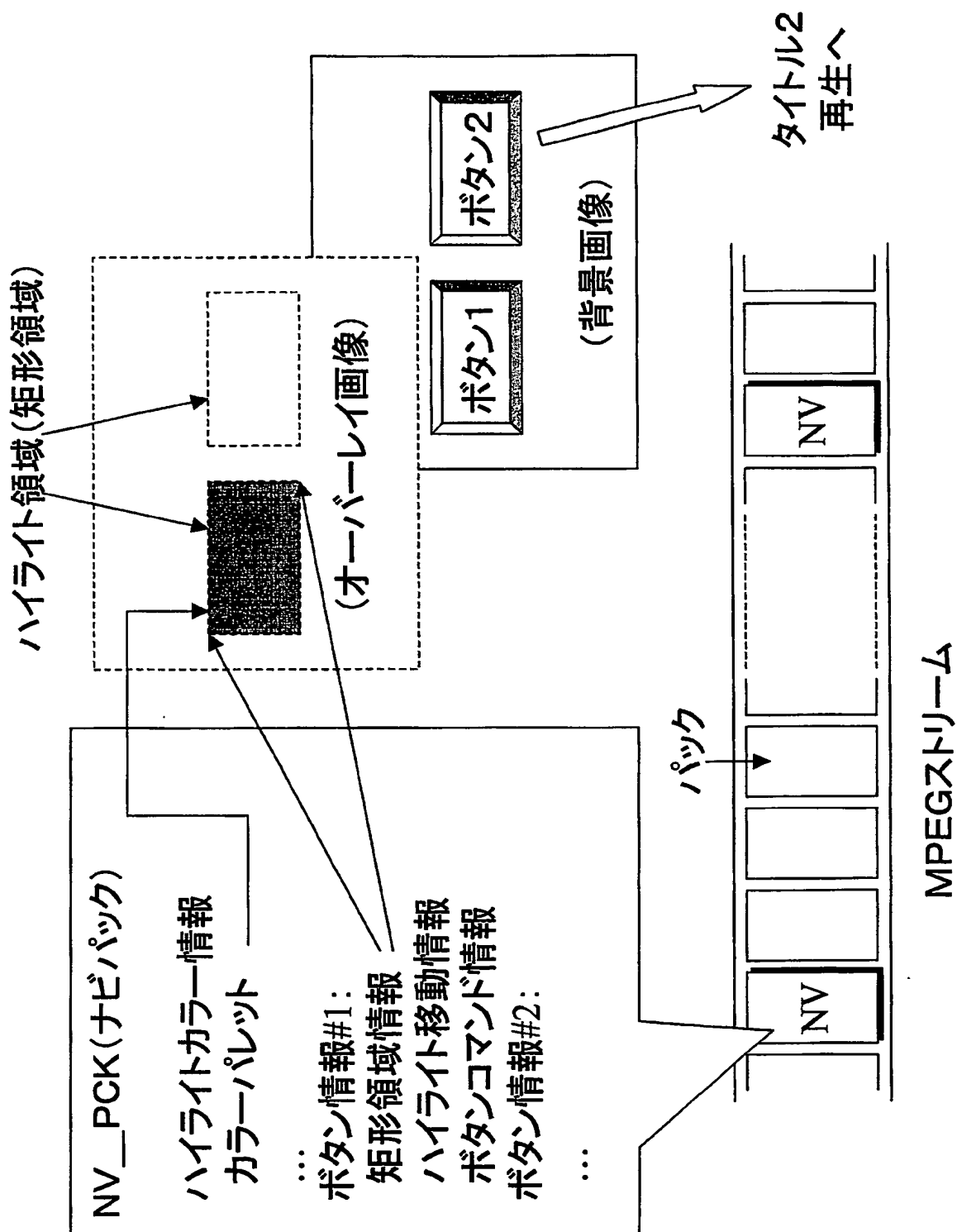
- S 1 1 0 1 プレイリスト再生開始ステップ
- S 1 1 0 2 プレイリスト終了判定ステップ
- S 1 1 0 3 タイムイベント時刻判定ステップ
- S 1 1 0 4 タイムイベント生成ステップ
- S 1 1 0 5 イベントハンドラ処理ステップ

- S 1 2 0 1 プレイリスト再生開始ステップ
- S 1 2 0 2 モード切替判定ステップ
- S 1 2 0 3 現有効モード判定ステップ
- S 1 2 0 4 J a v a (登録商標) モードへの切り替えステップ
- S 1 2 0 5 DVD 互換モードへの切り替えステップ
- S 4 9 0 1 判定開始ステップ
- S 4 9 0 2 読み込み必須データの読み込みステップ
- S 4 9 0 3 読み込みオプションデータの優先度判定ステップ
- S 4 9 0 4 ローカルメモリの空き容量とデータサイズの比較ステップ
- S 4 9 0 5 オプションデータの読み込みステップ
- S 4 9 0 6 判定終了ステップ

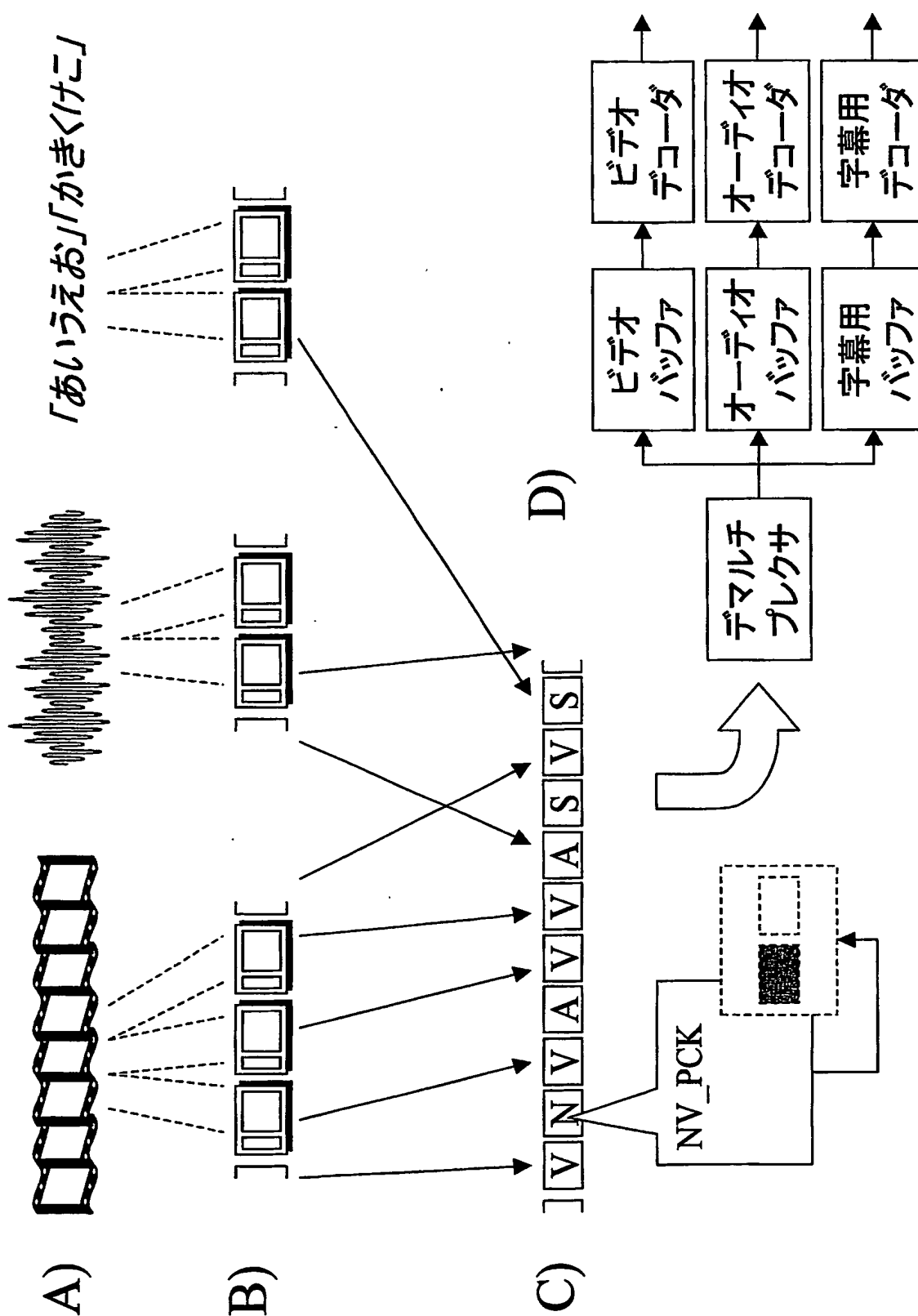
【書類名】 図面
【図 1】



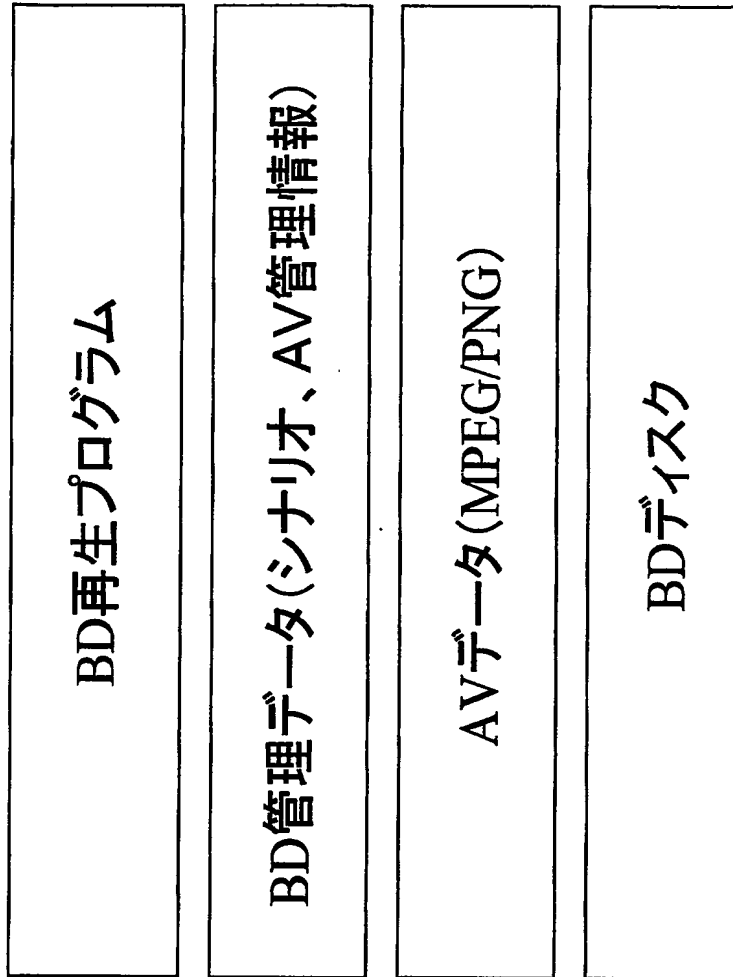
【図 2】



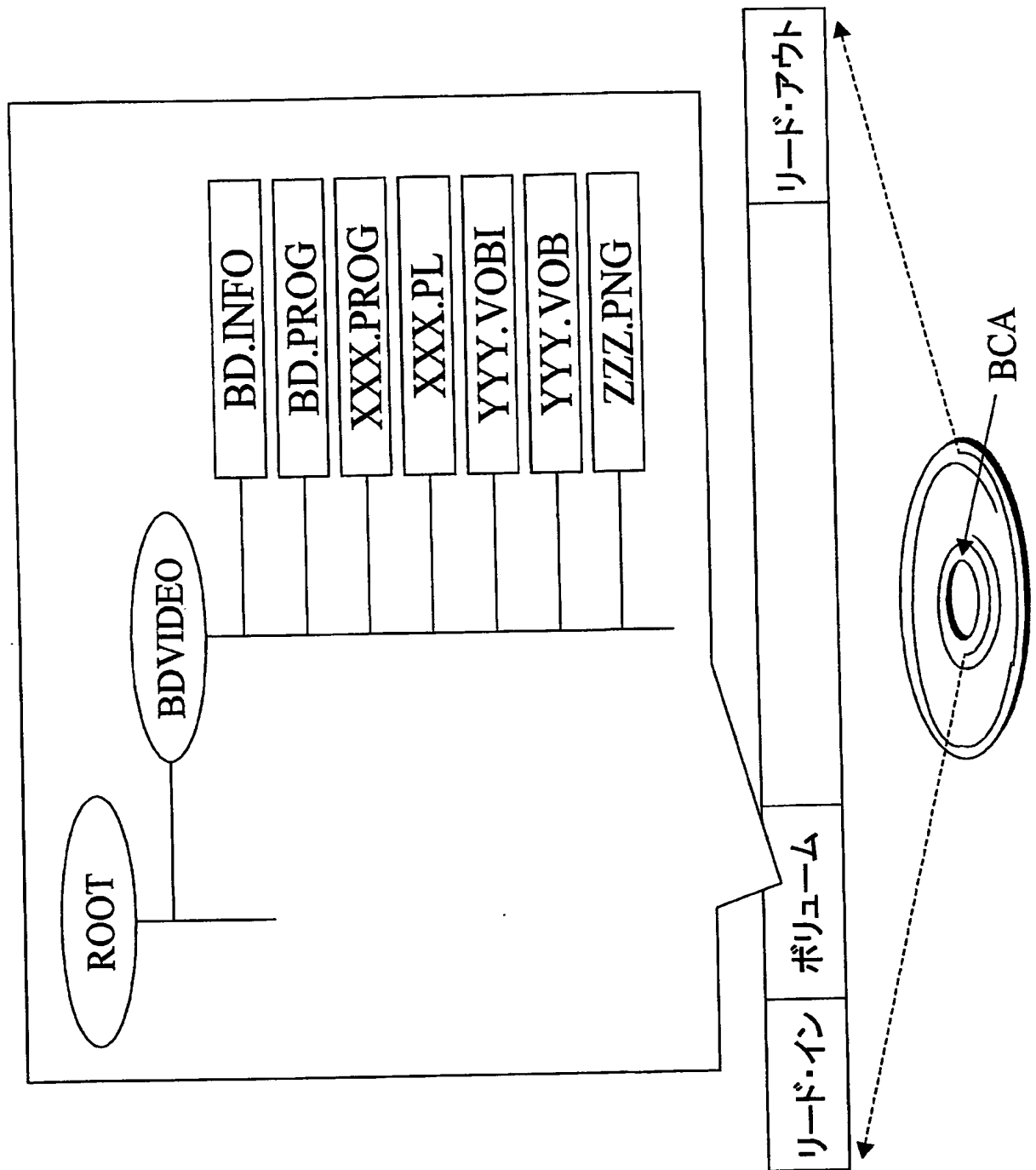
【図 3】



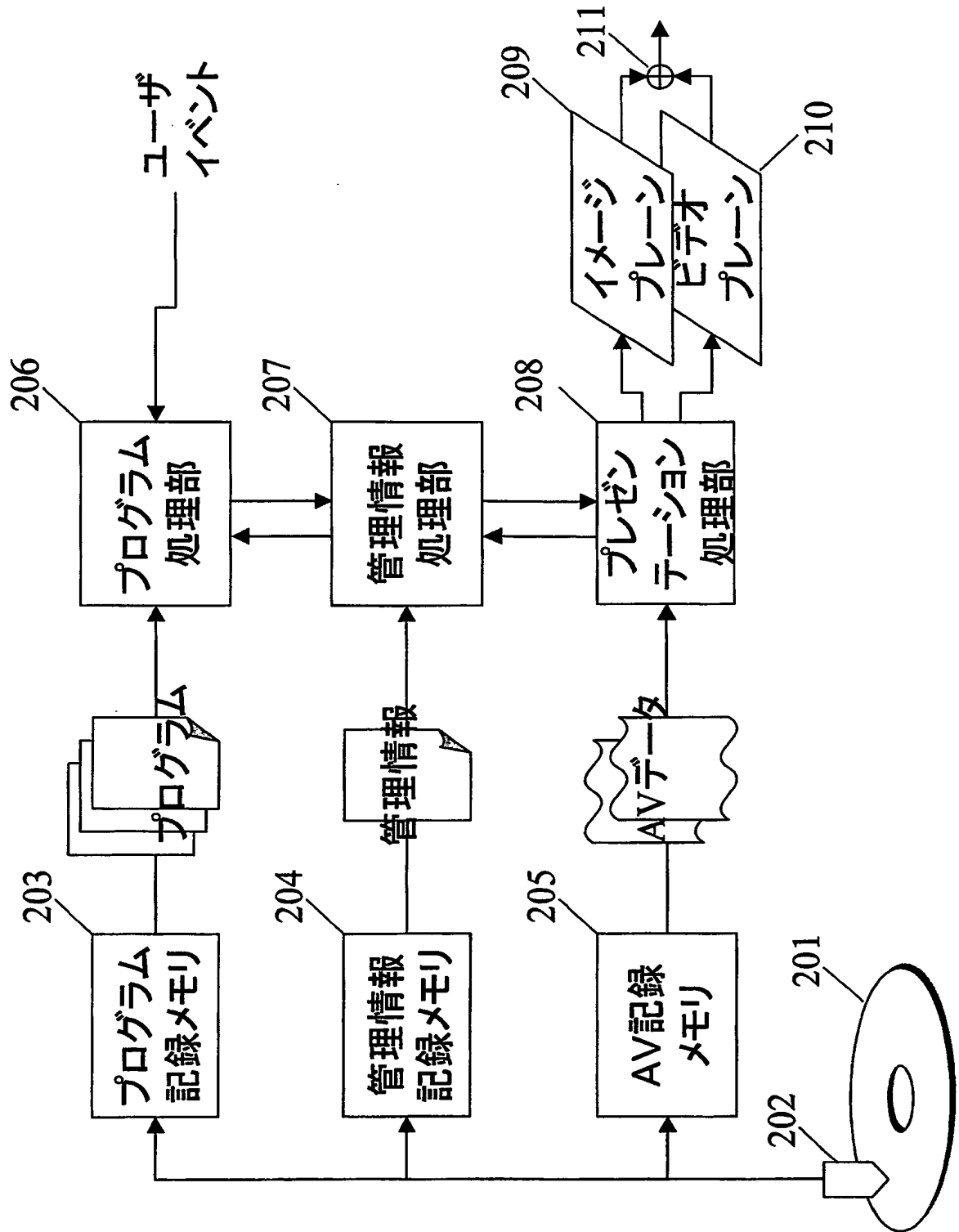
【図 4】



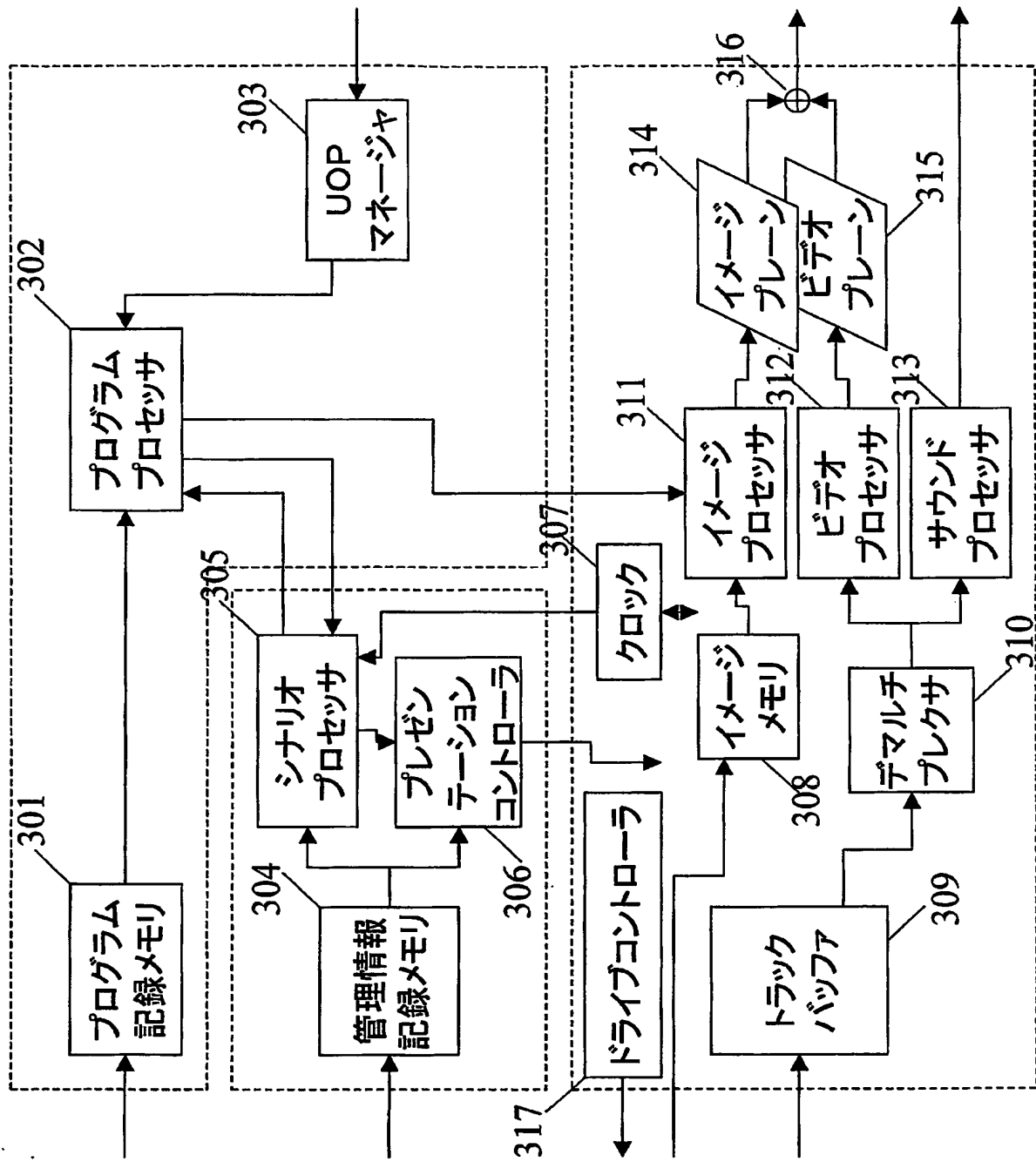
【図 5】



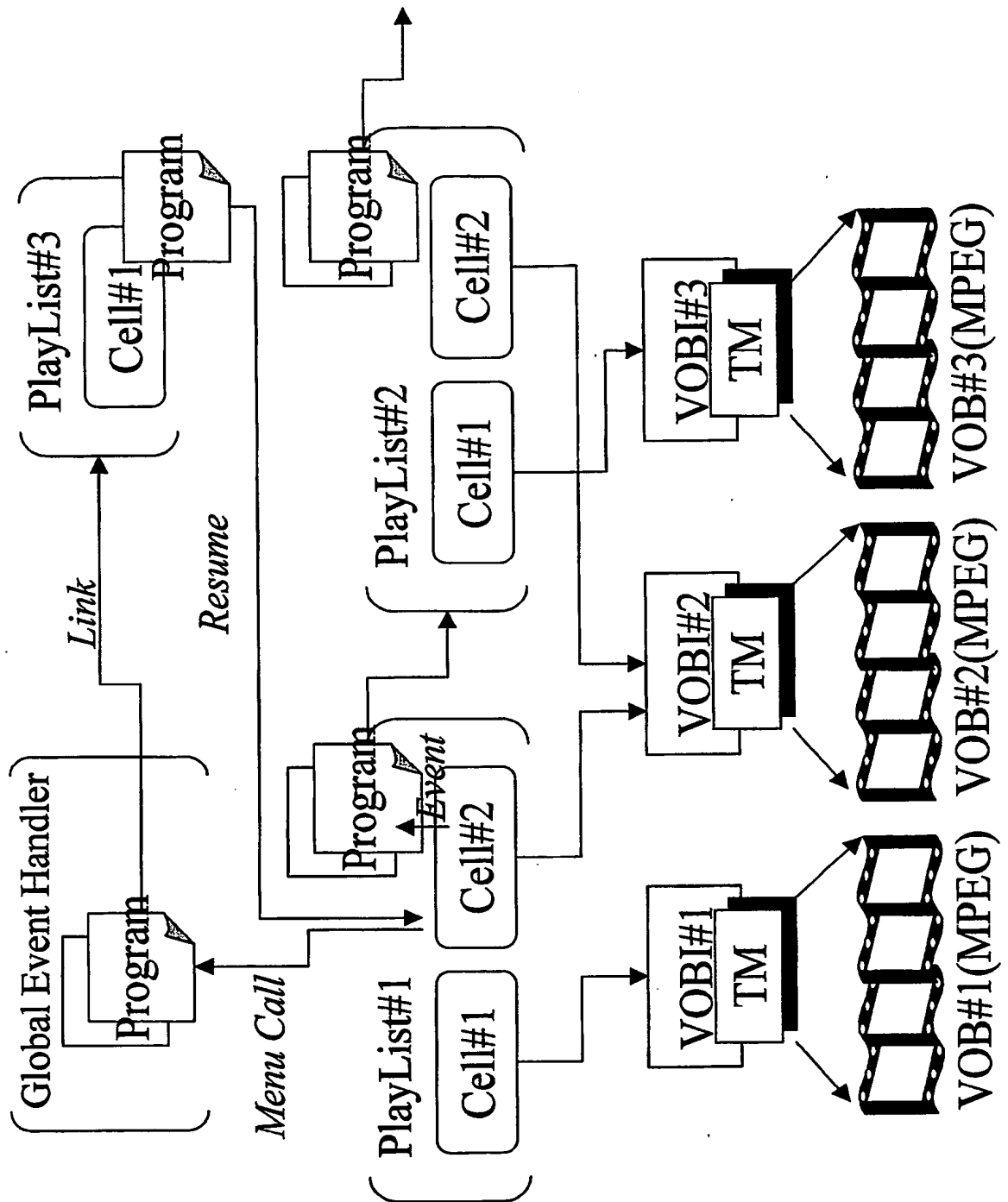
【図 6】



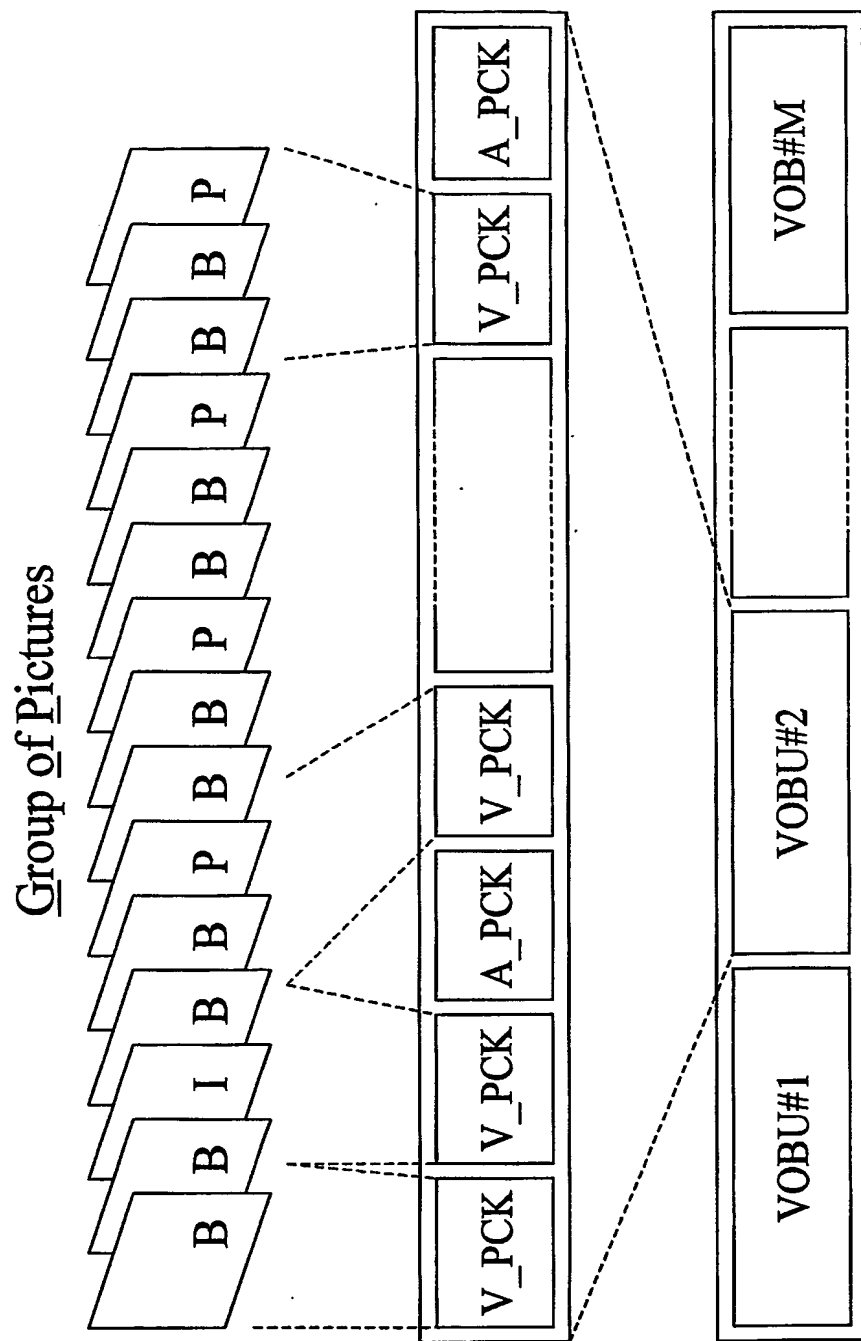
【図 7】



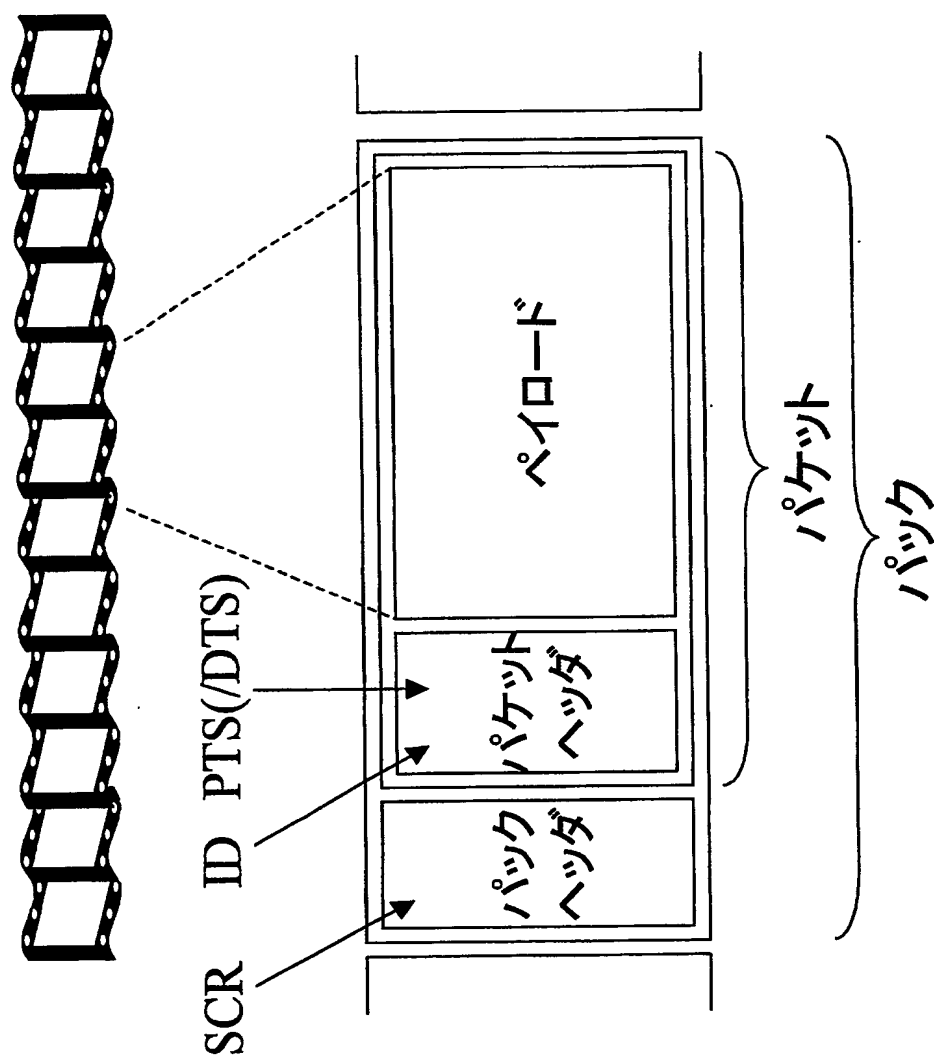
【図 8】



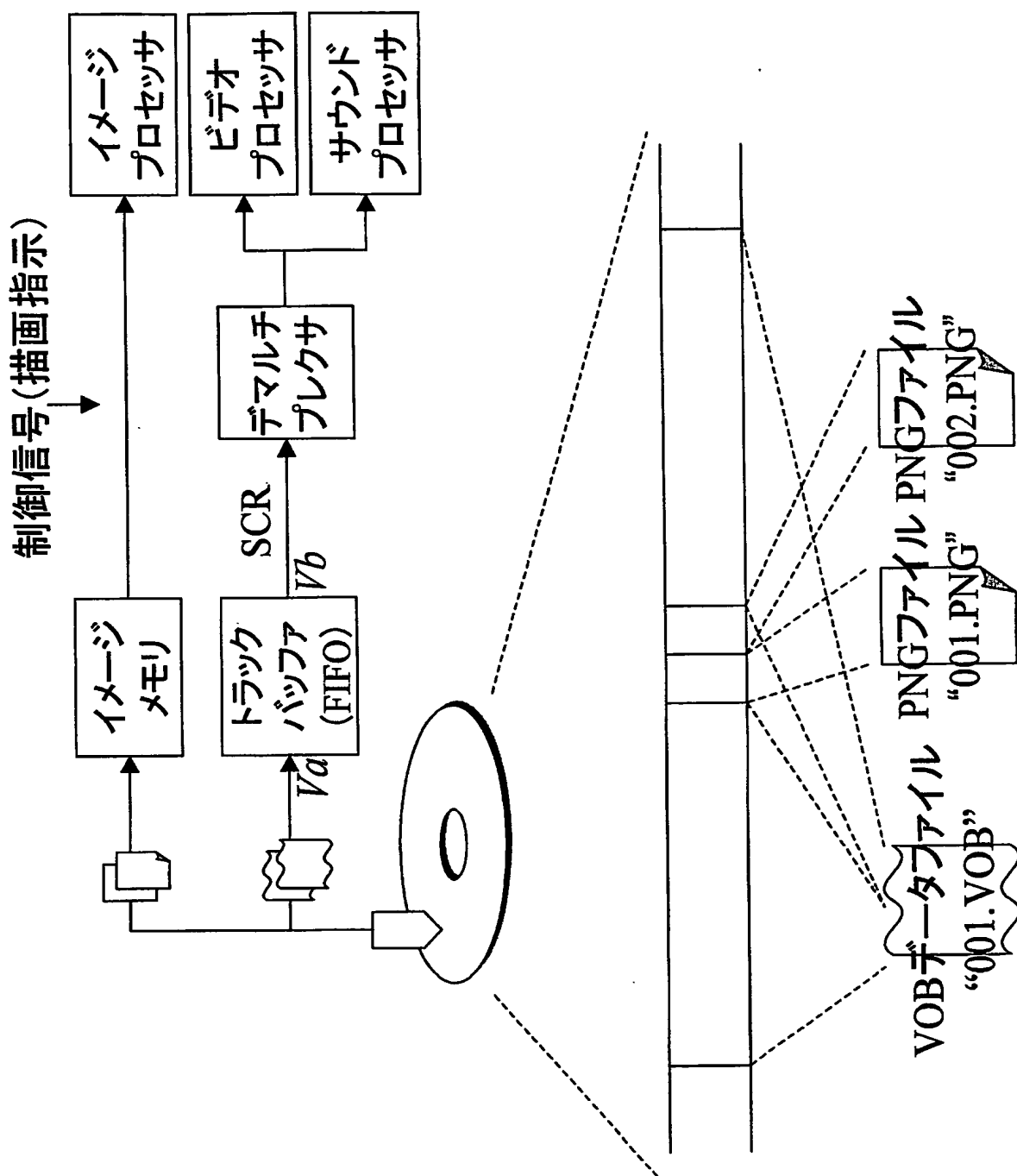
【図 9】



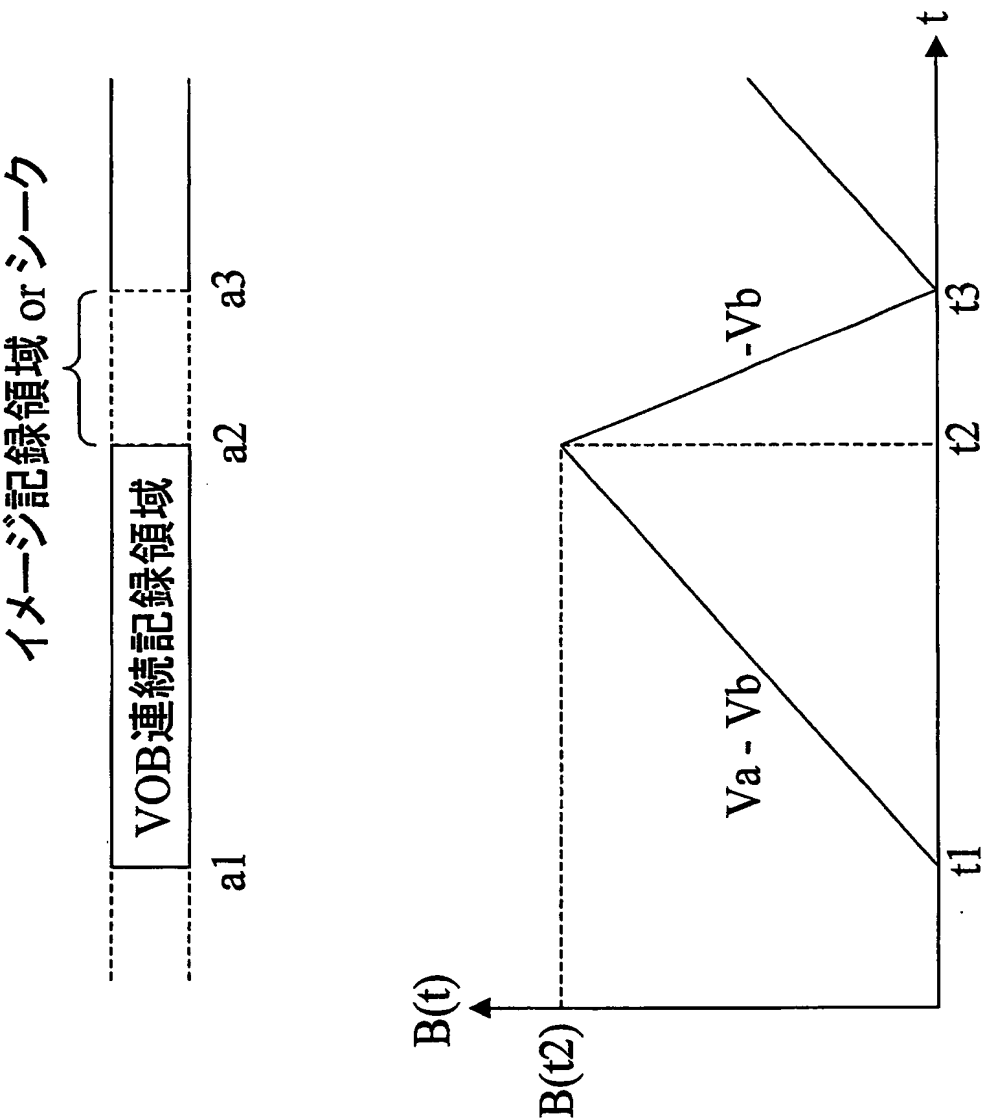
【図 10】



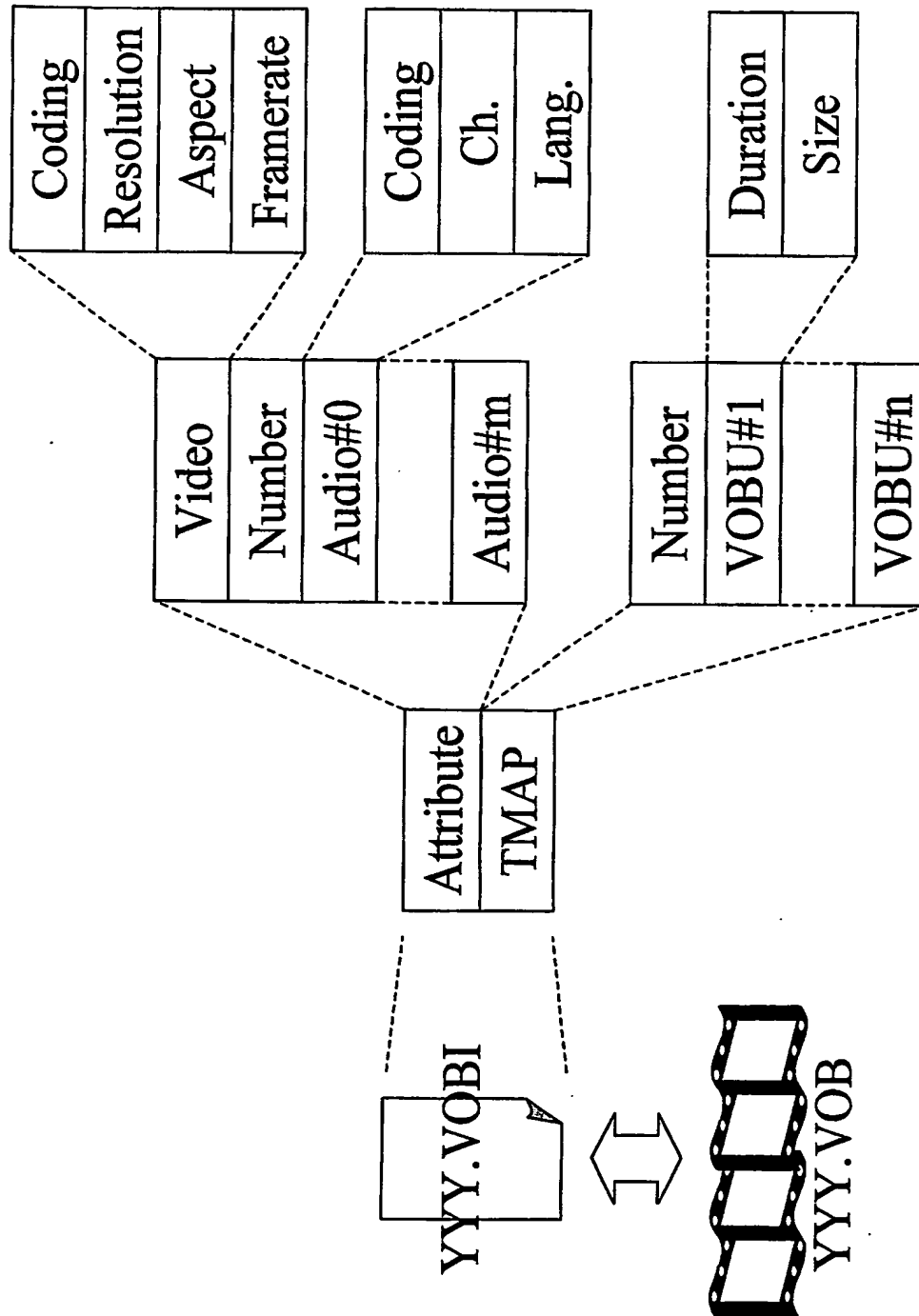
【図 1 1】



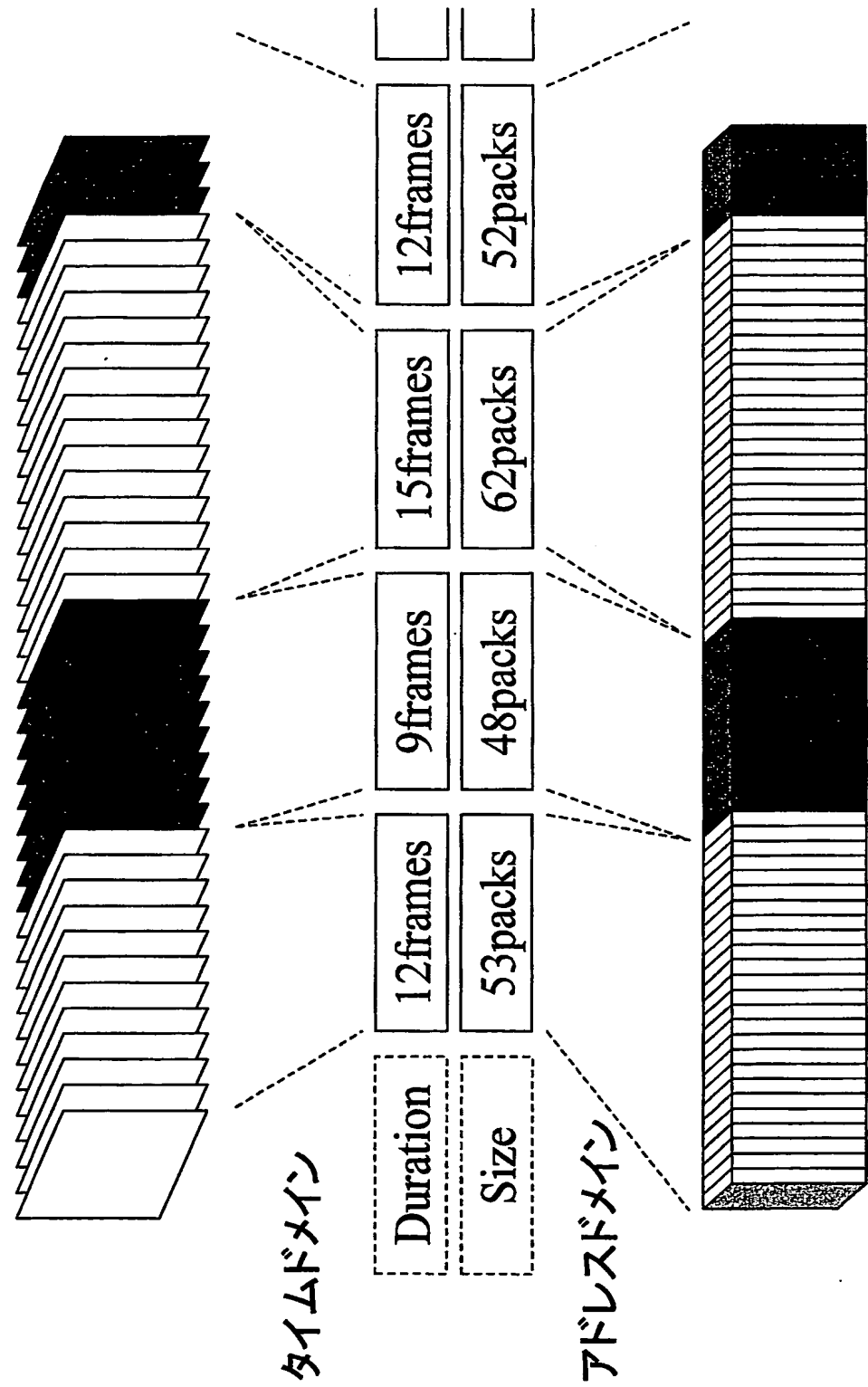
【図 12】



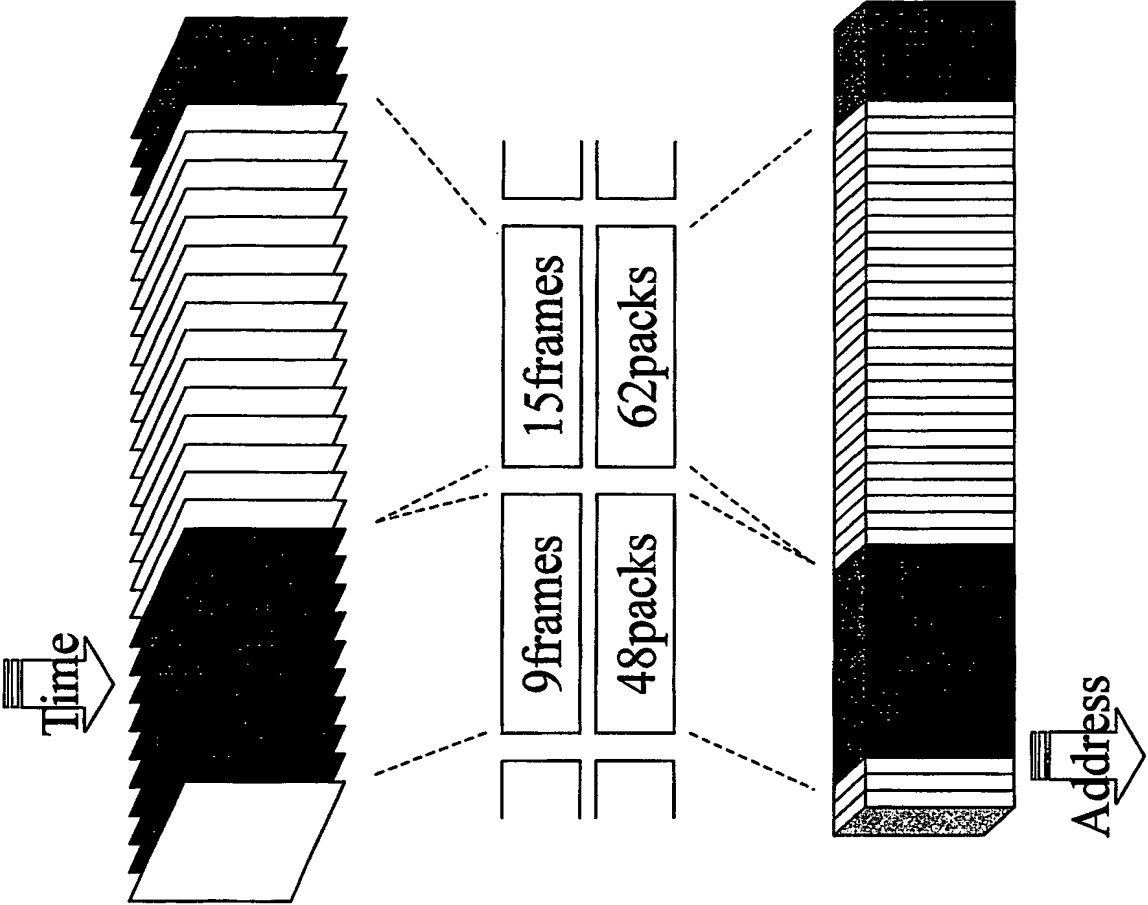
【図 13】



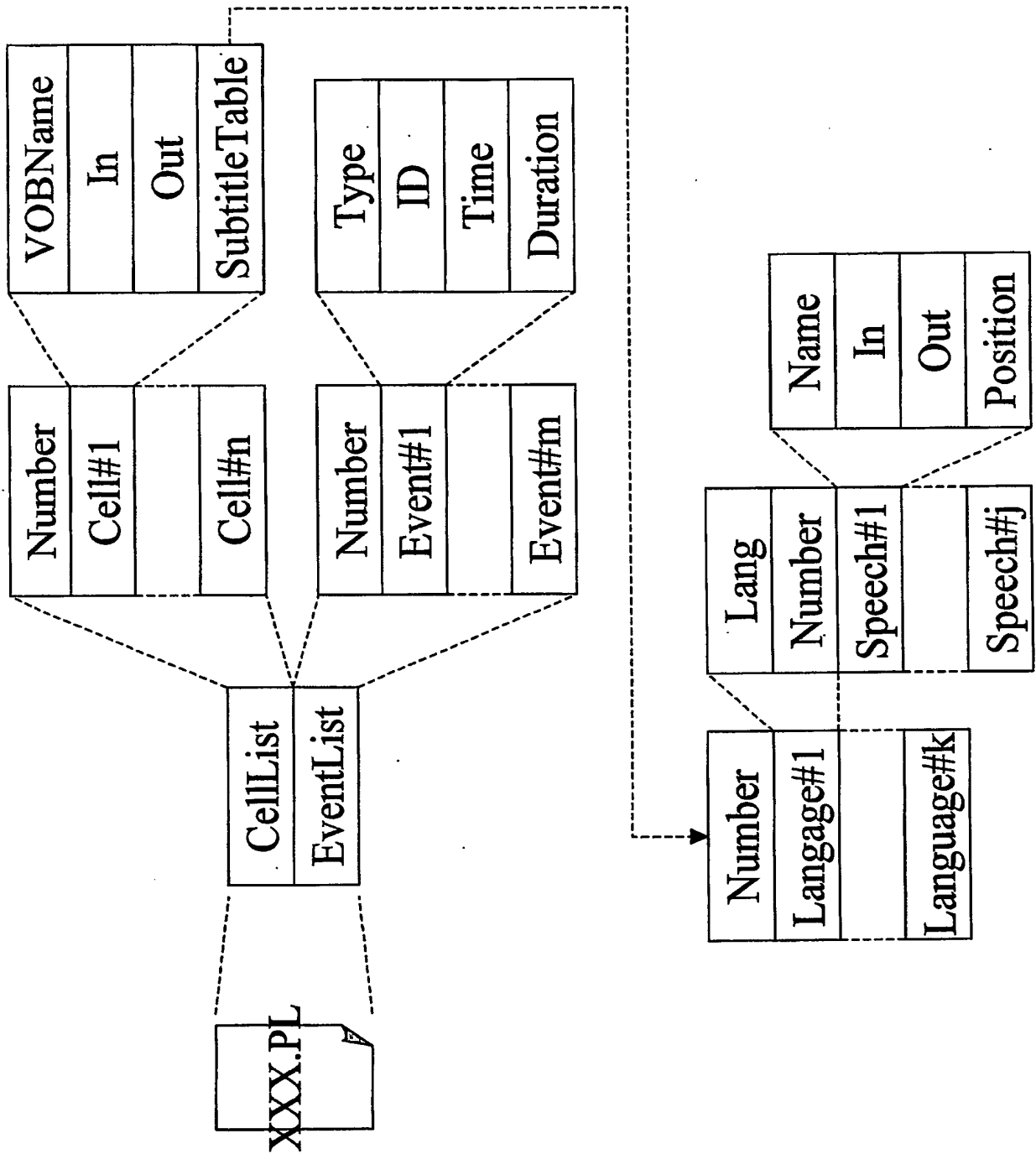
【図 14】



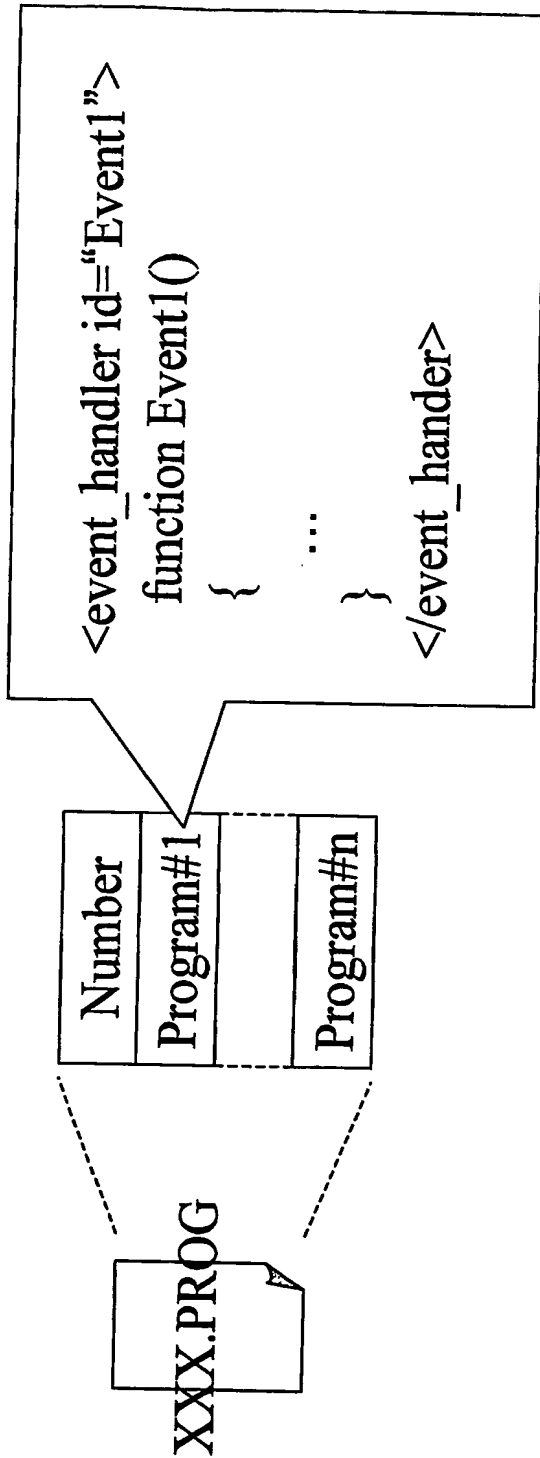
【図 15】



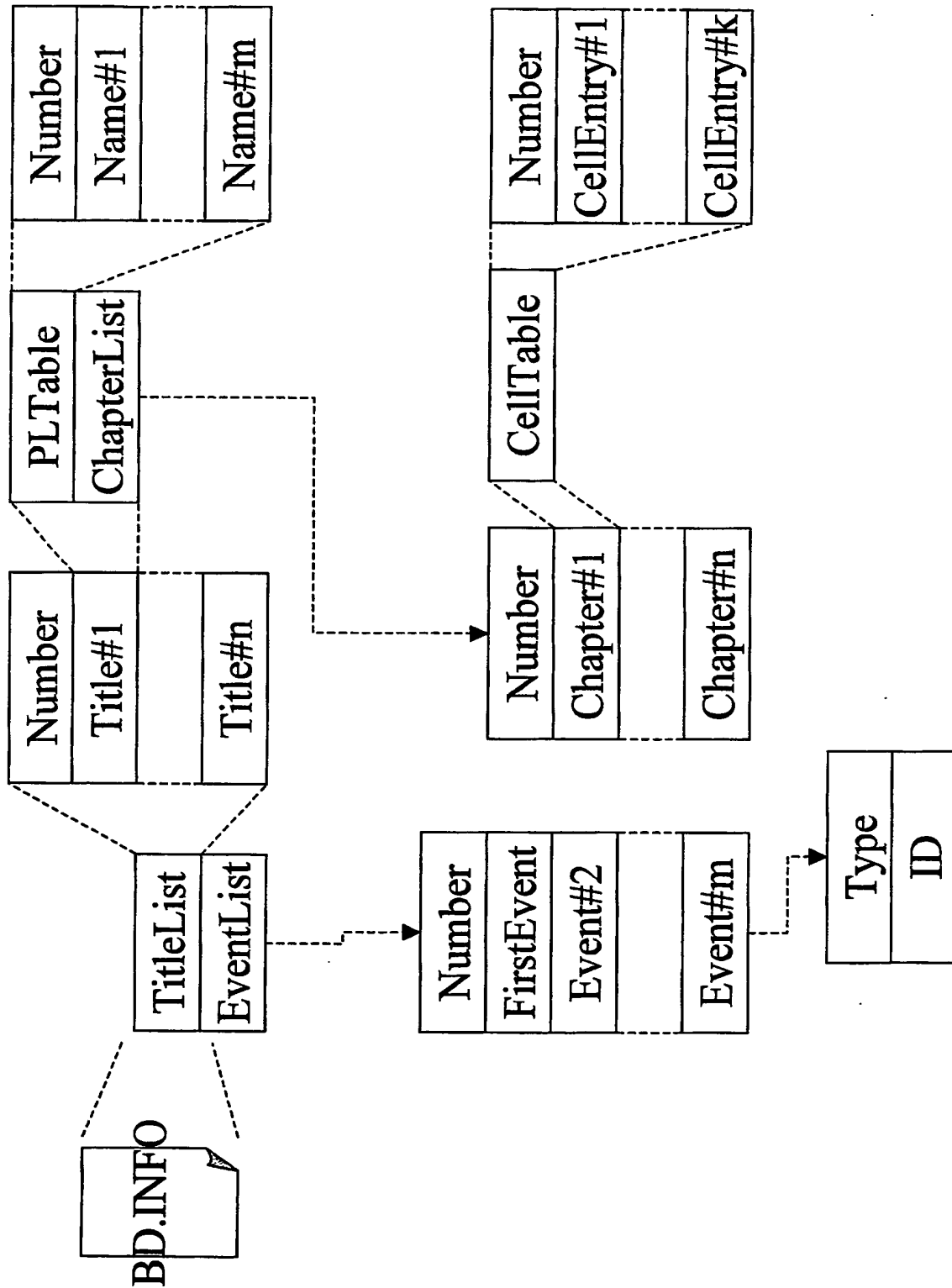
【図 16】



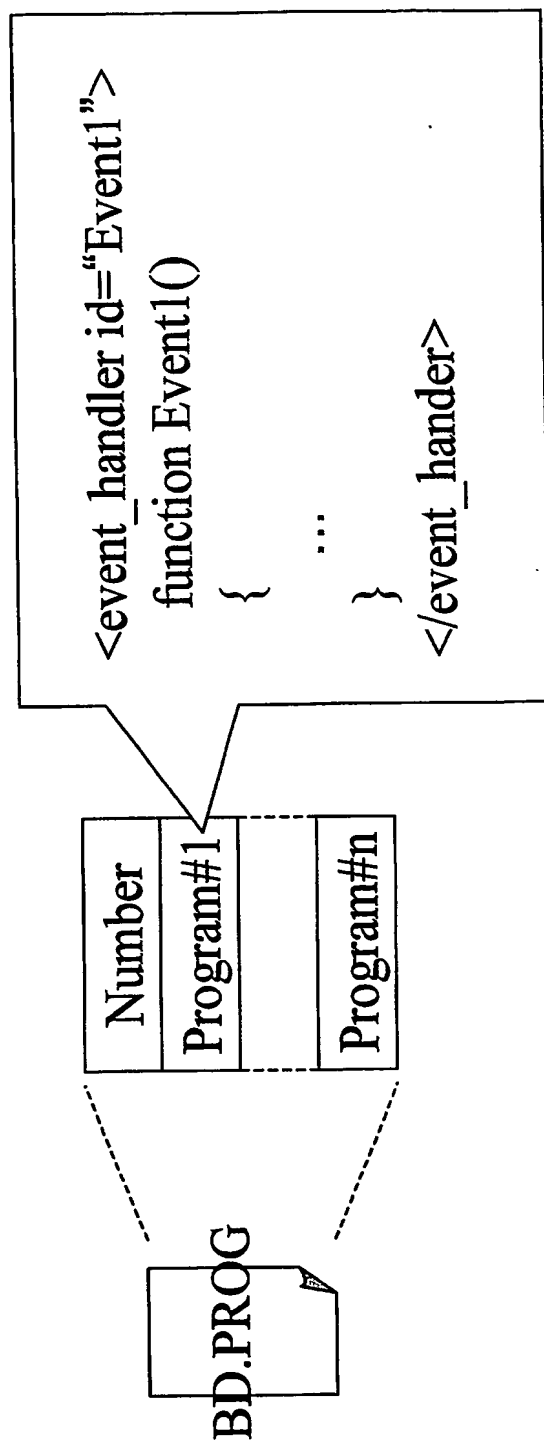
【図 17】



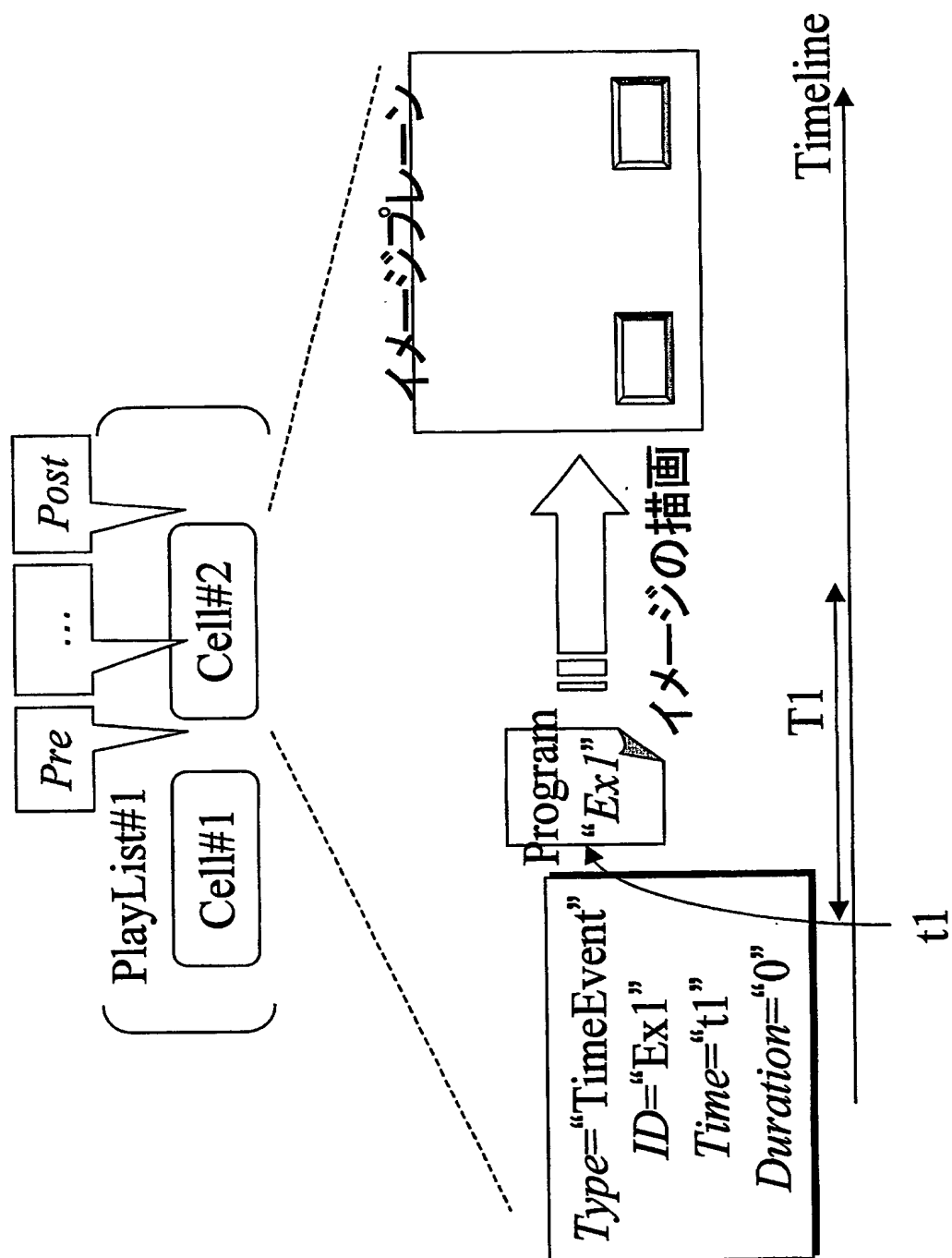
【図 18】



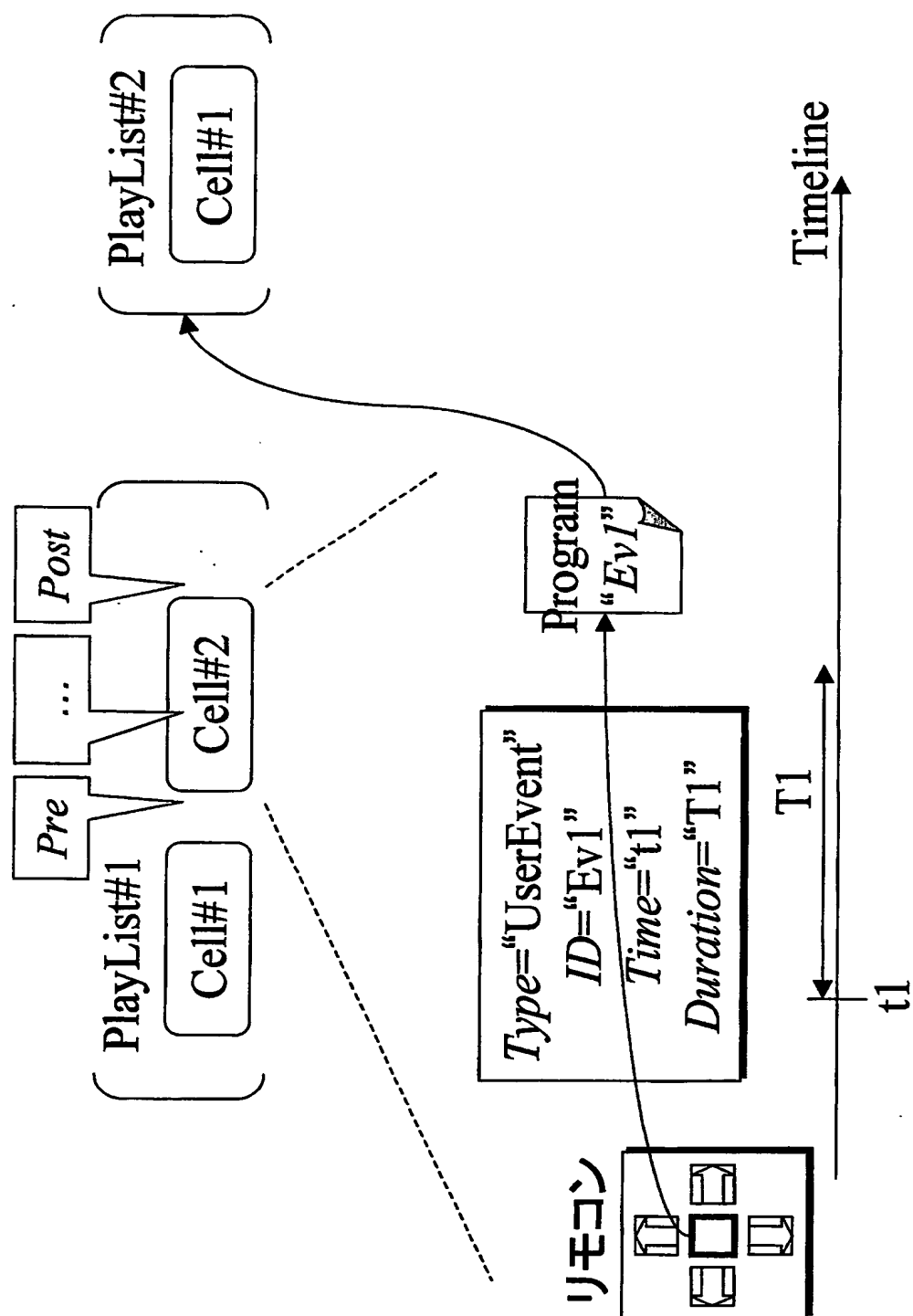
【図 19】



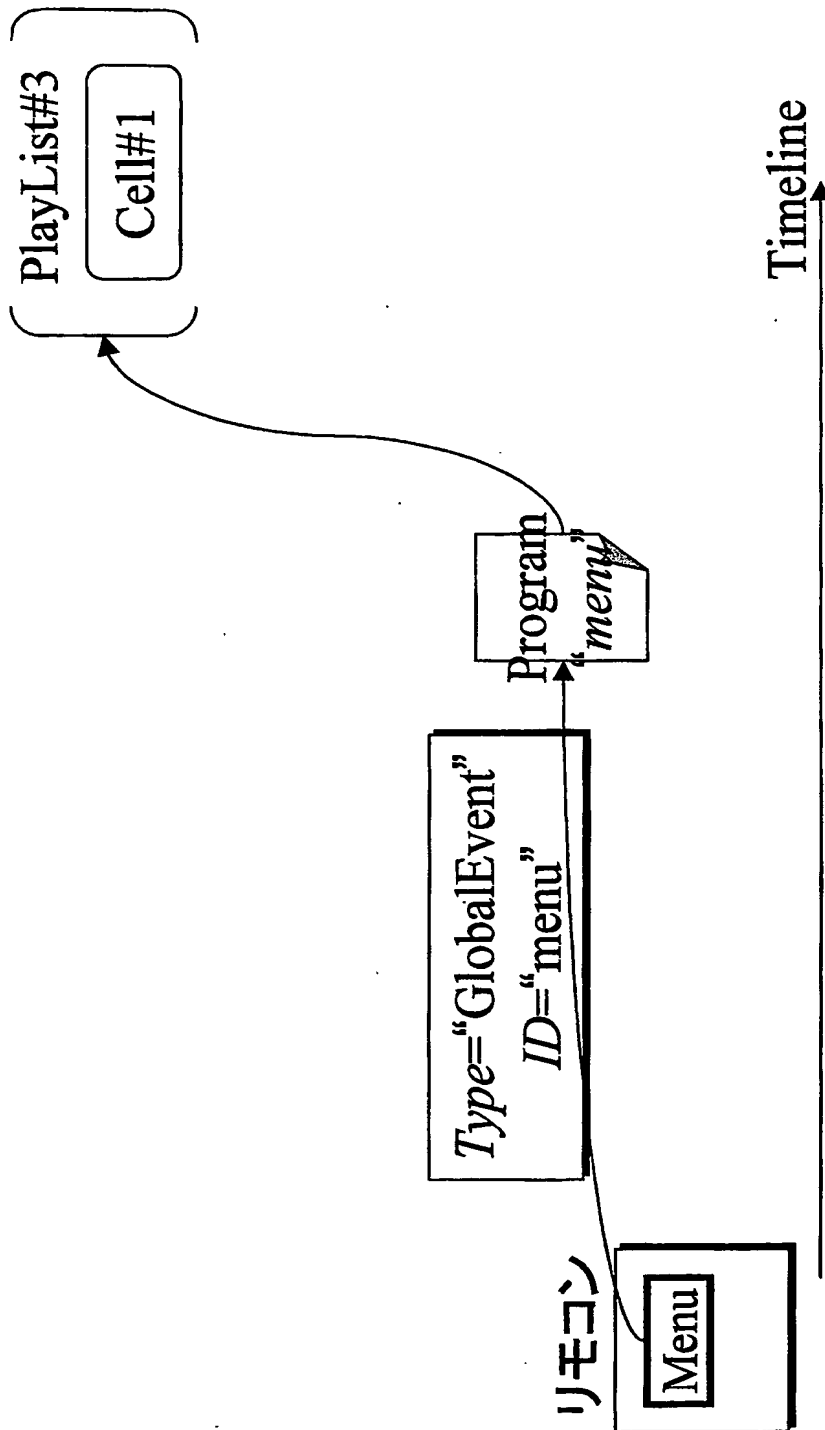
【図 20】



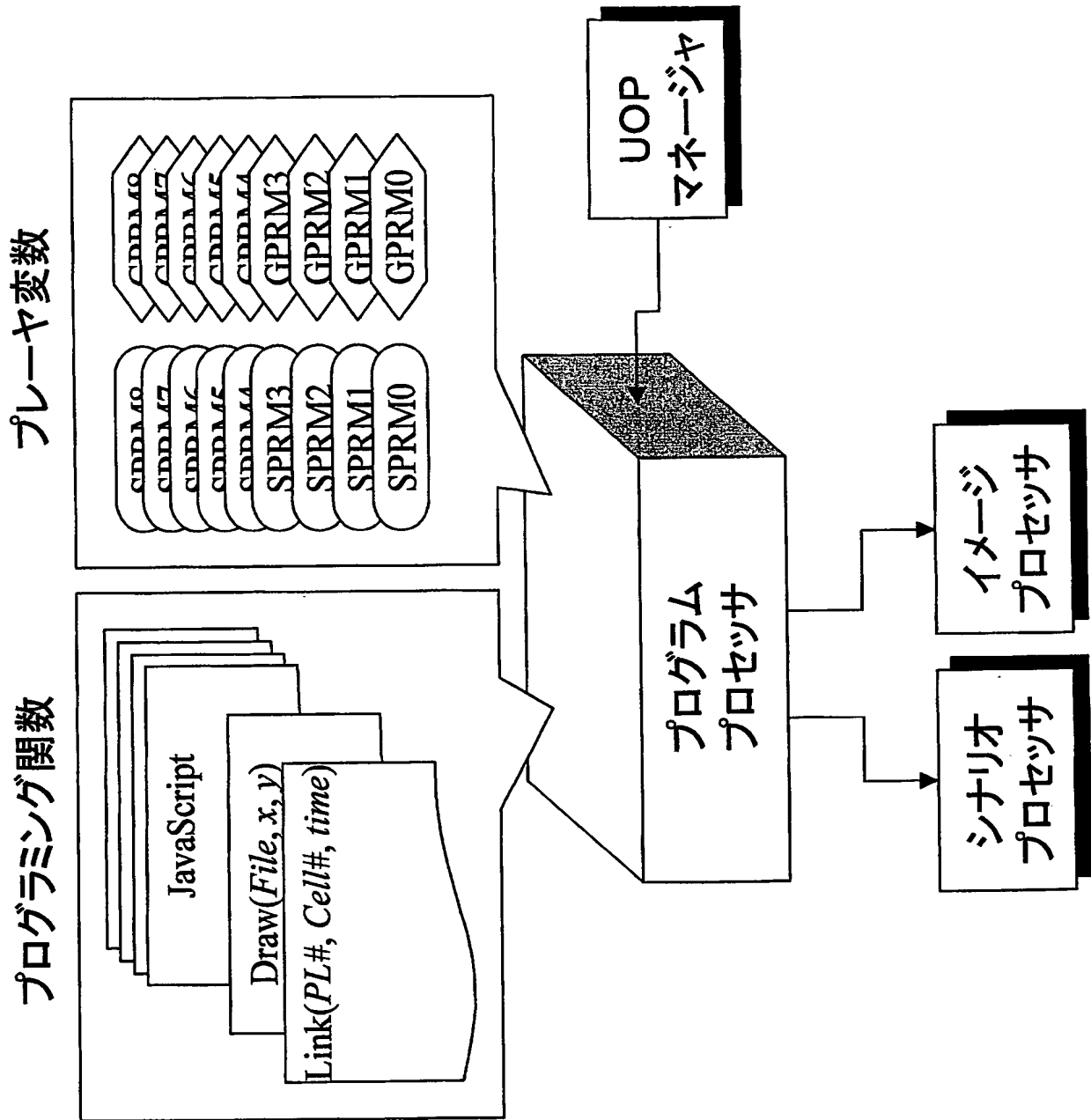
【図 21】



【図 22】



【図 23】



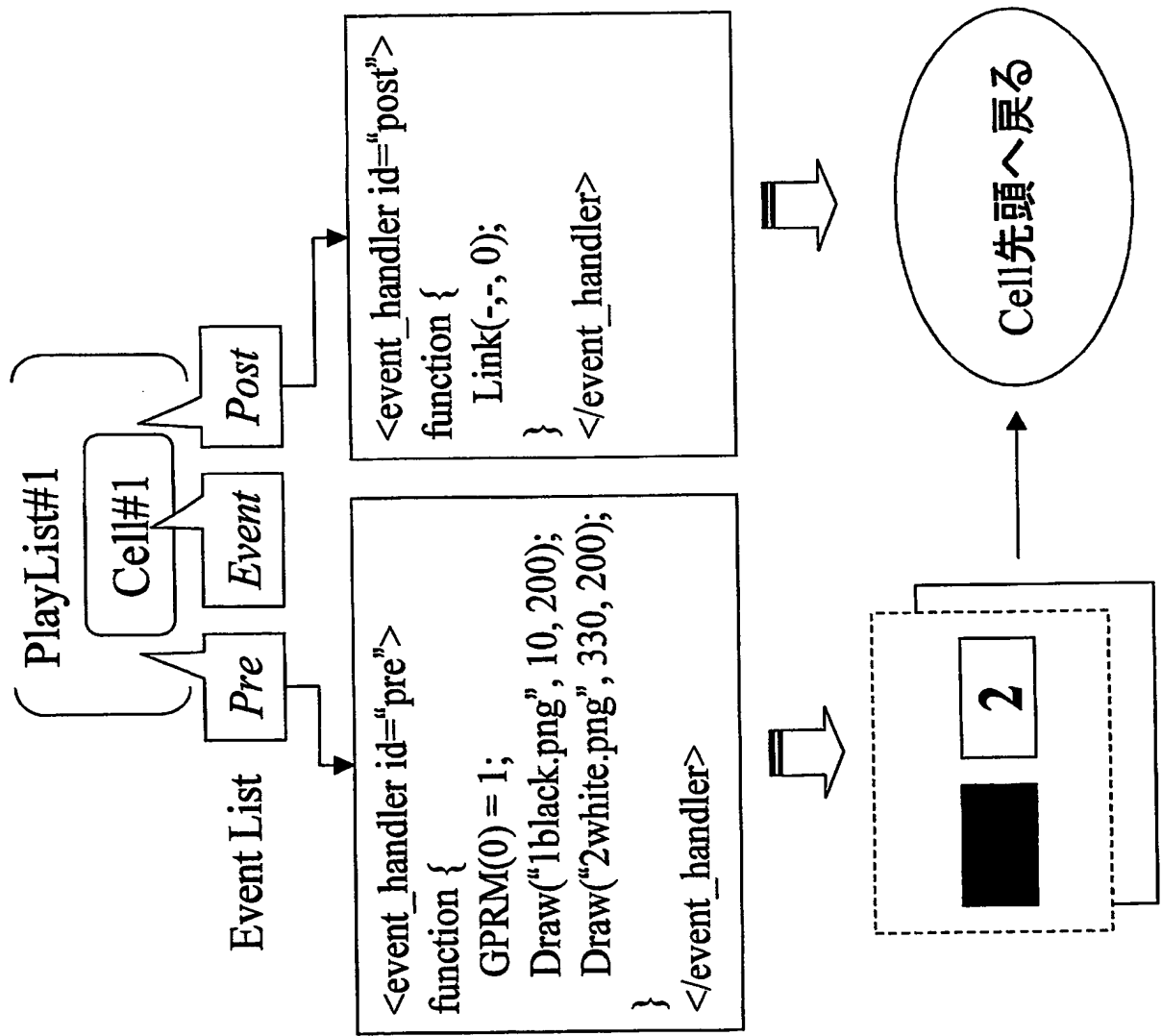


【図 24】

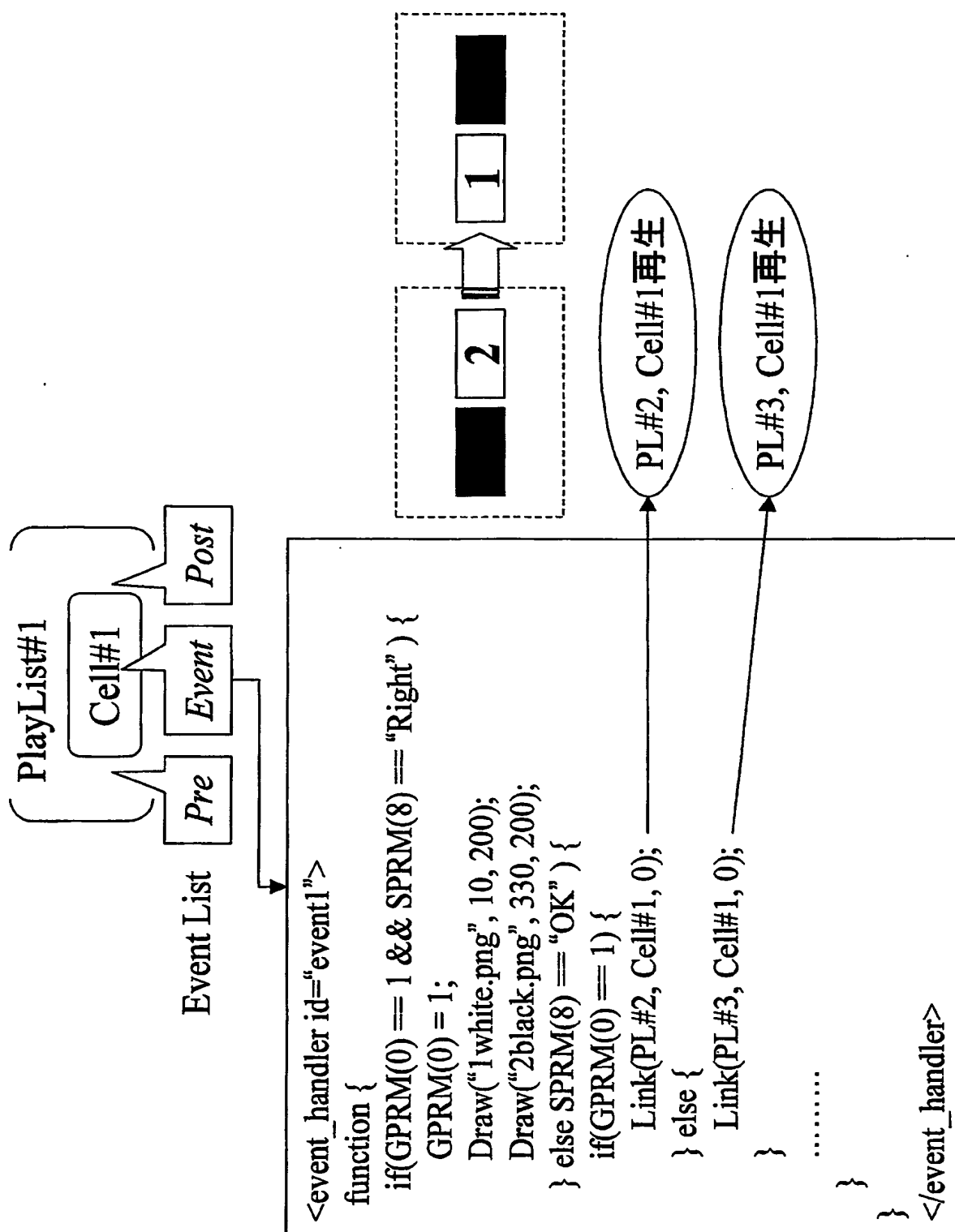
プレーヤ変数(システムパラメータ)

0	Language Code	11	Player audio mixing mode for Karaoke	22	reserved
1	Audio stream number	12	Country code for parental management	23	Player status
2	Subtitle stream number	13	Parental level	24	reserved
3	Angle number	14	Player configuration for Video	25	reserved
4	Title number	15	Player configuration for Audio	26	reserved
5	Chapter number	16	Language code for AST	27	reserved
6	Program number	17	Language code ext. for AST	28	reserved
7	Cell number	18	Language code for STST	29	reserved
8	Key name	19	Language coded ext. for STST	30	reserved
9	Navigation timer	20	Player region code	31	reserved
10	Current playback time	21	reserved	32	reserved

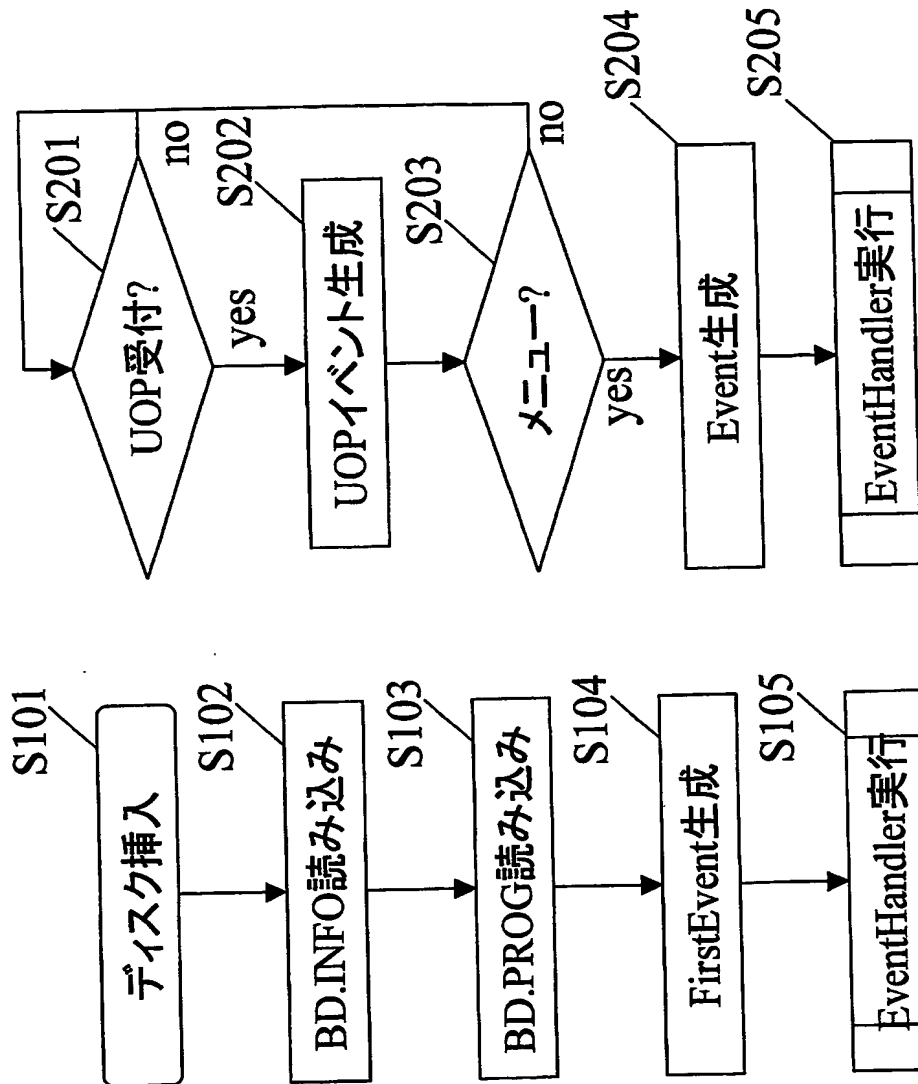
【図 25】



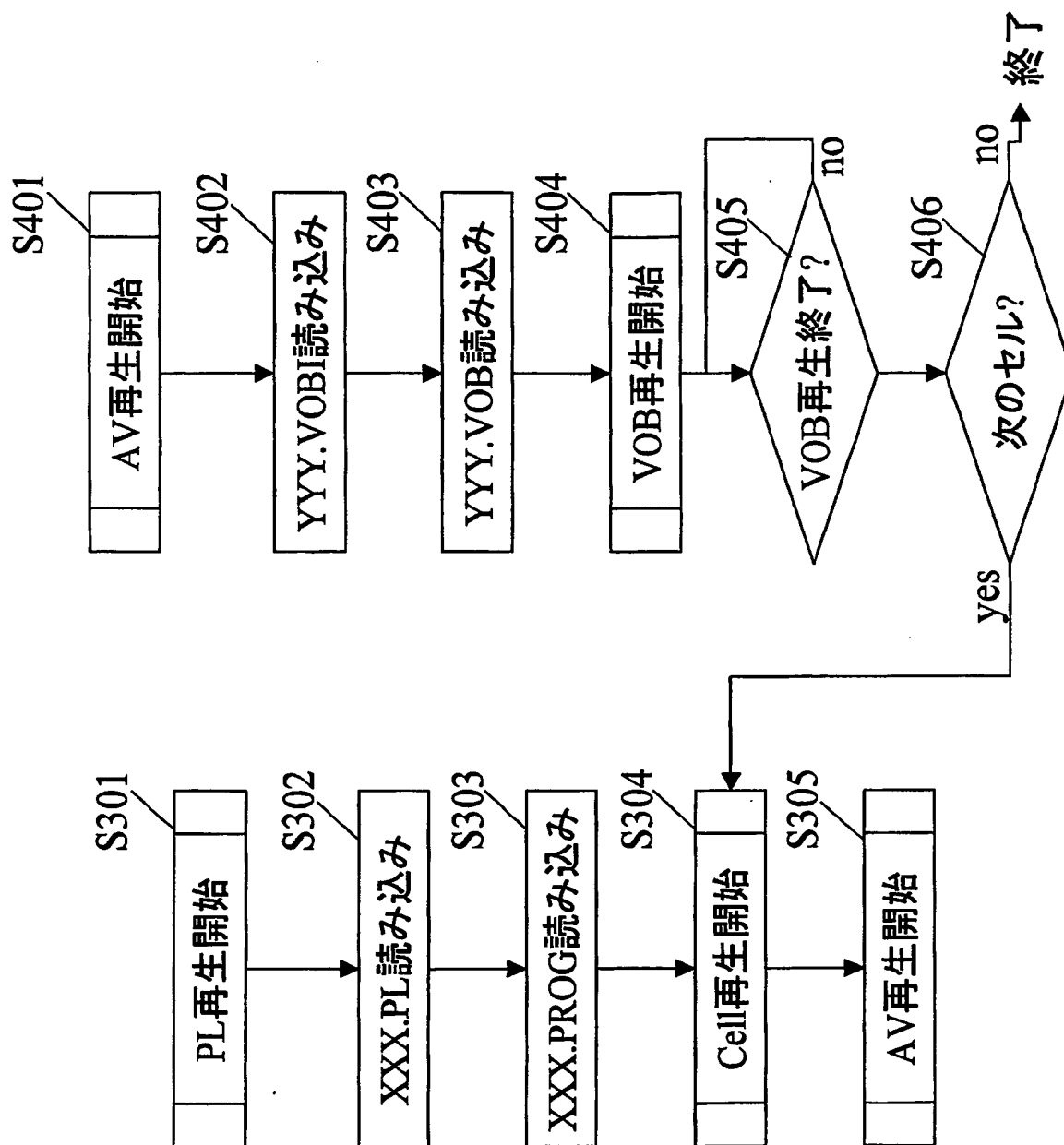
【図 26】



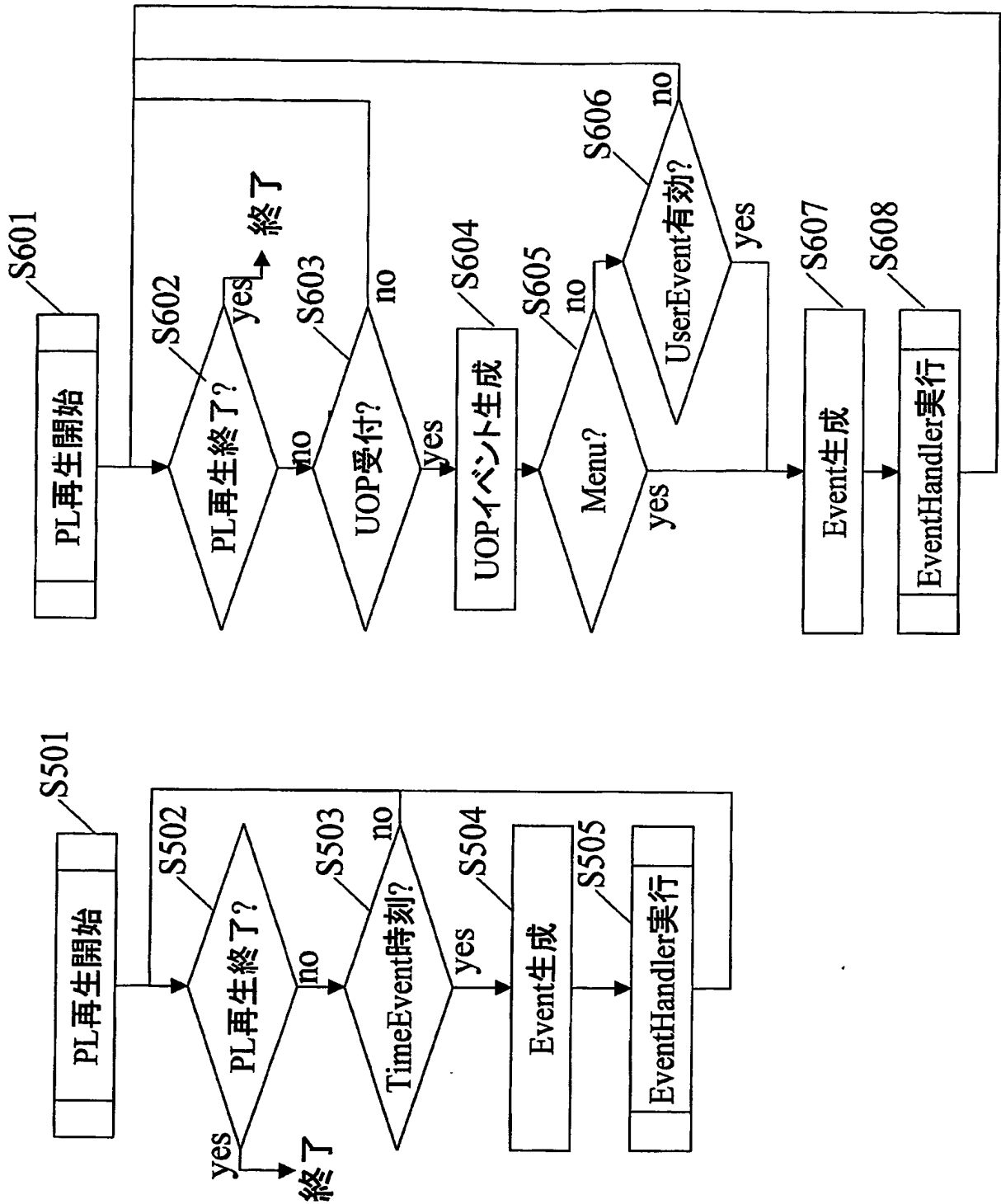
【図 27】



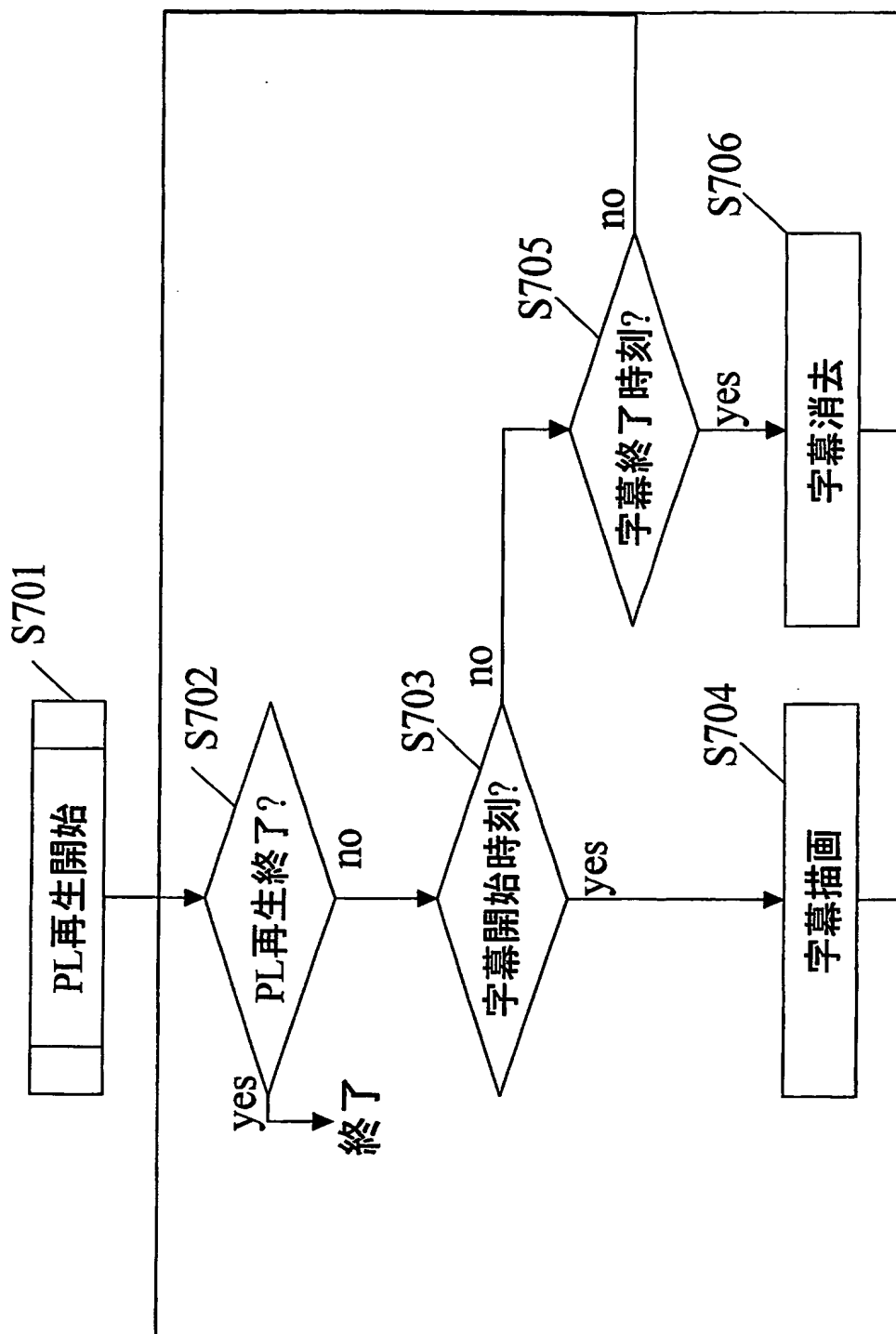
【図 28】



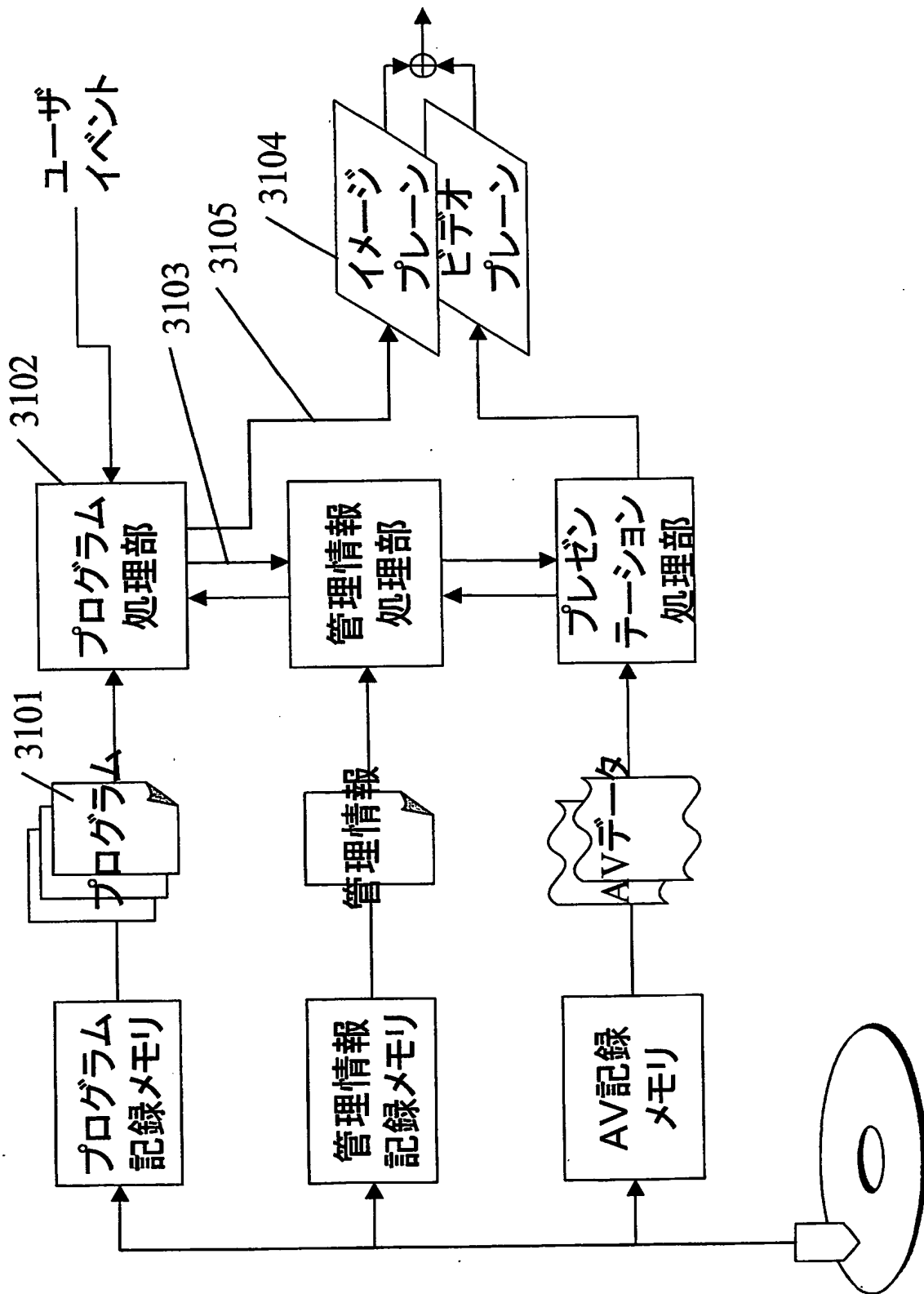
【図 29】



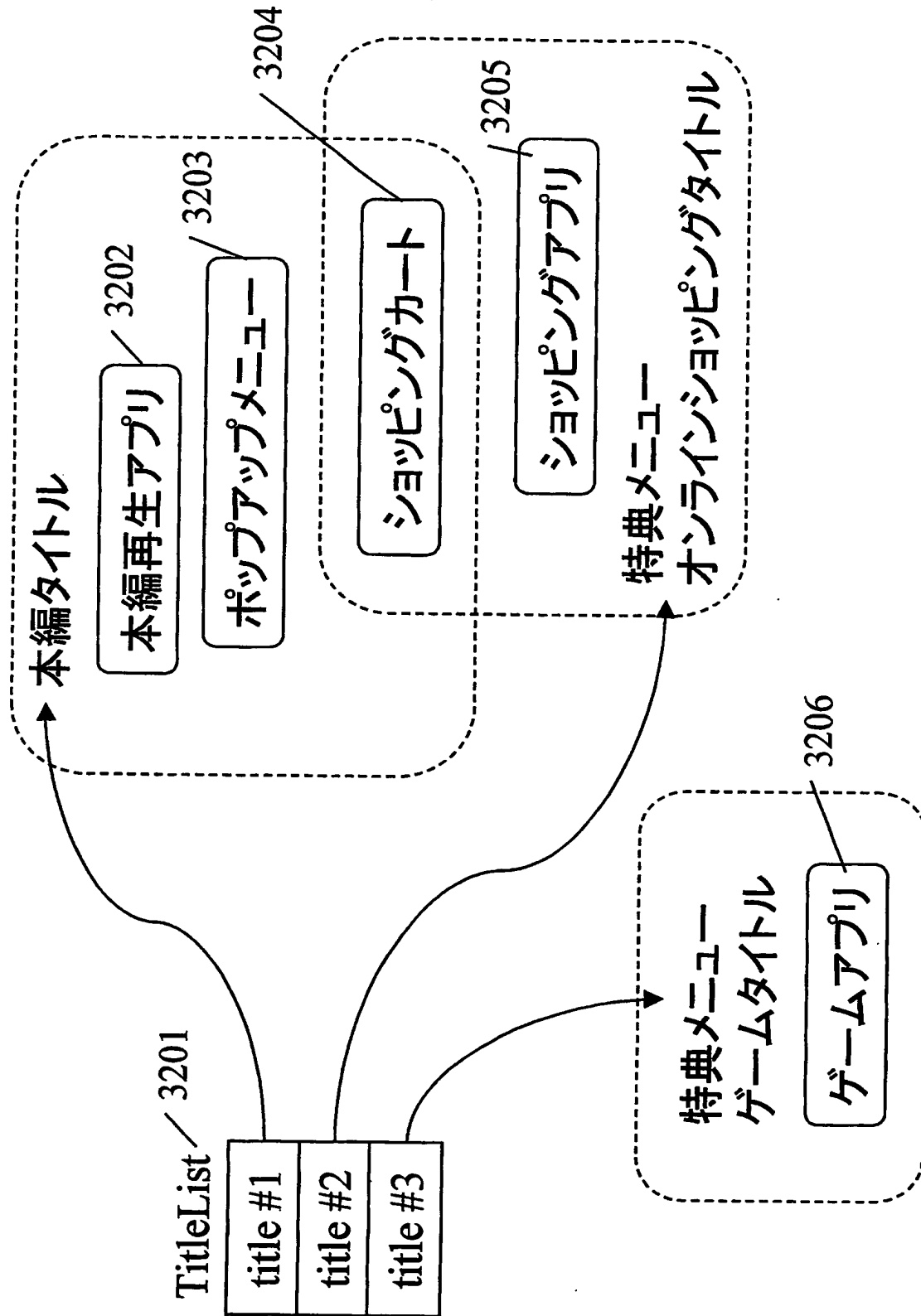
【図 30】



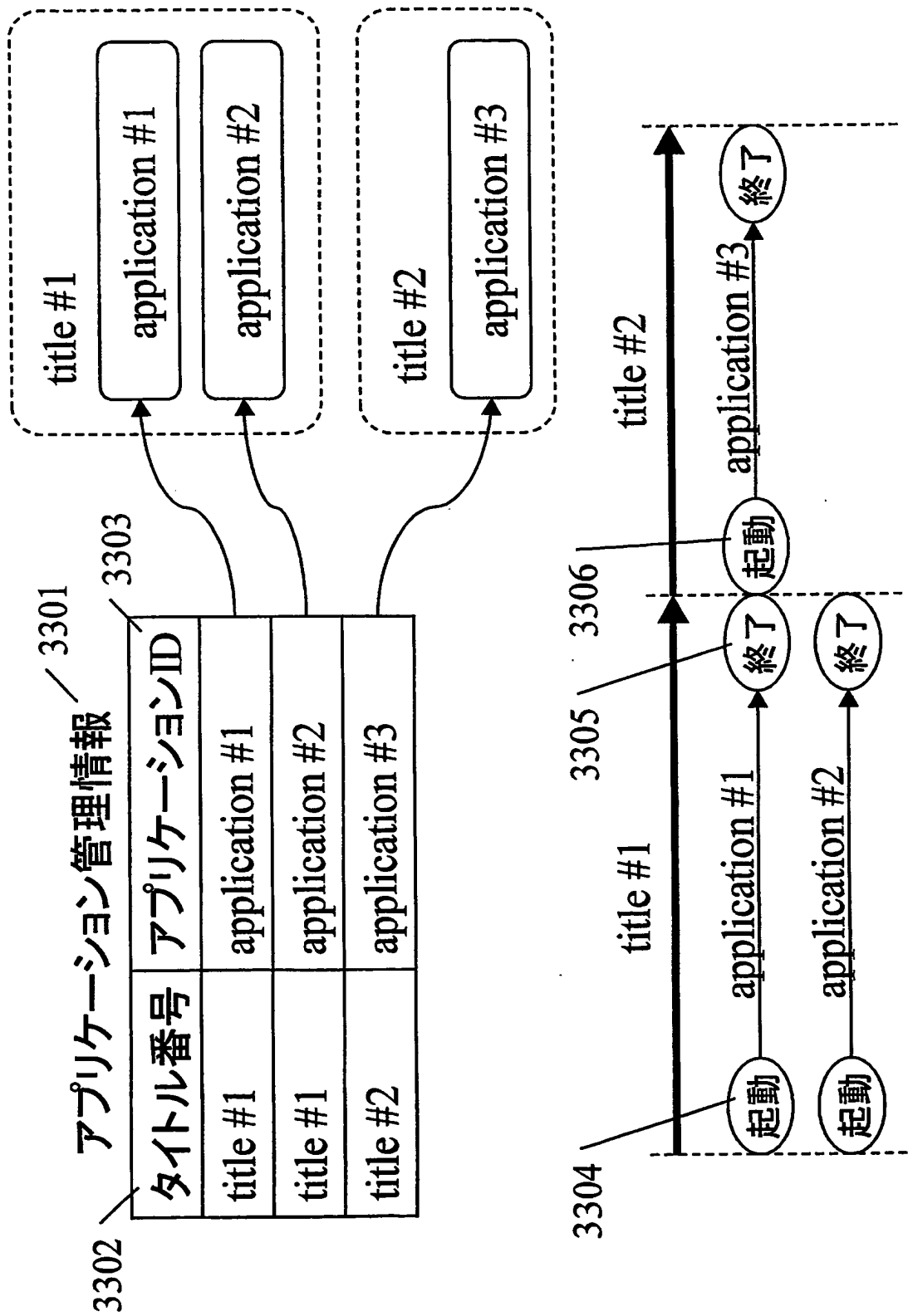
【図 31】



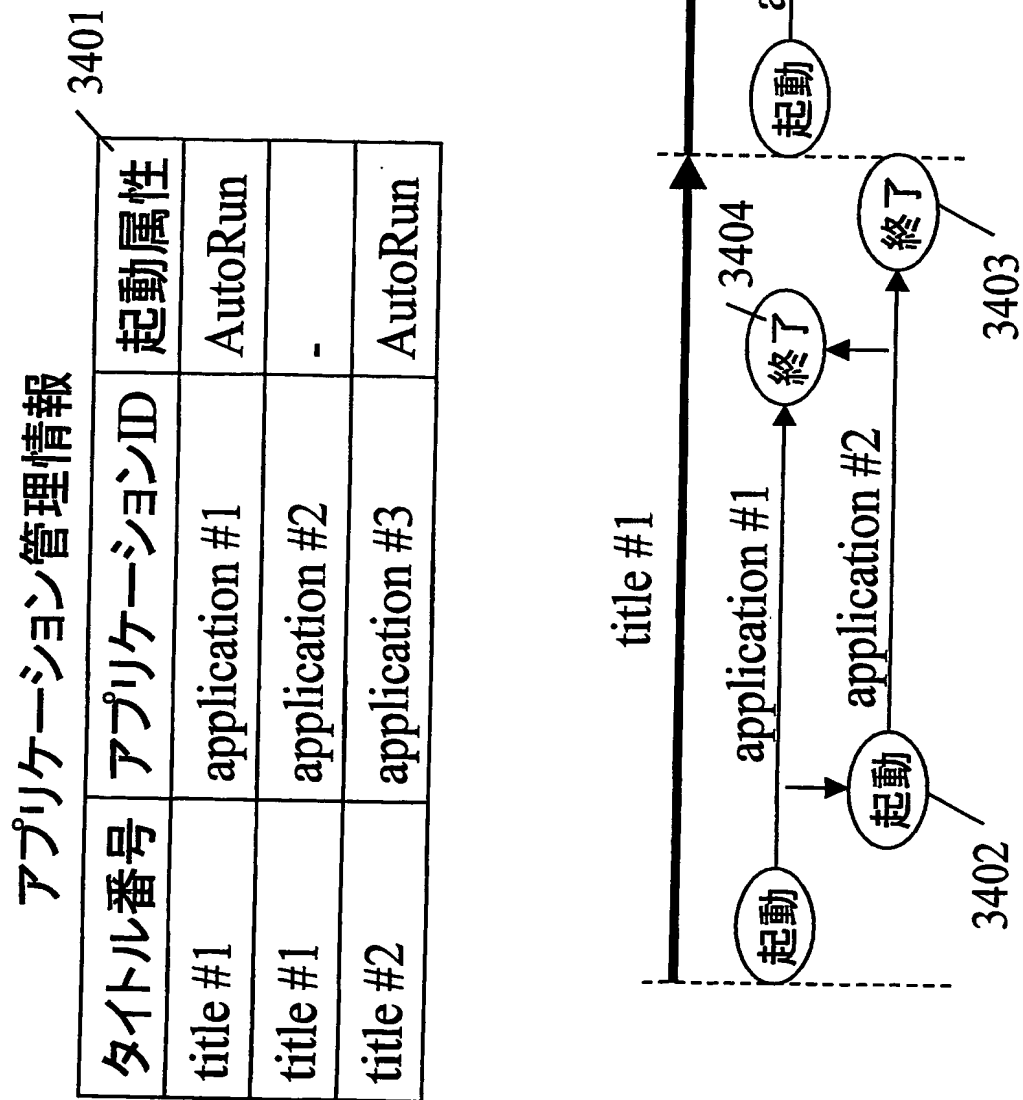
【図 32】



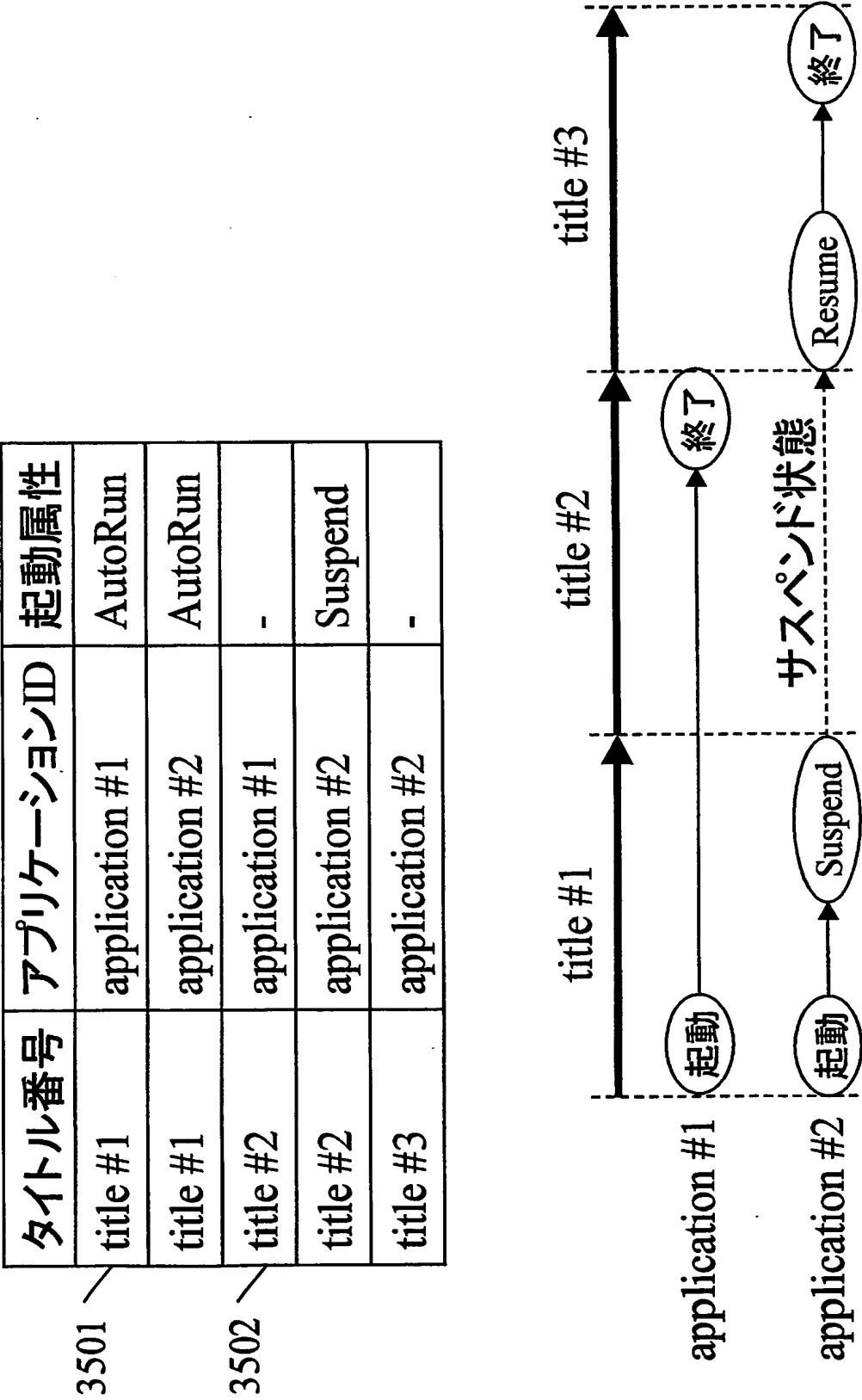
【図 33】



【図 34】



【図35】



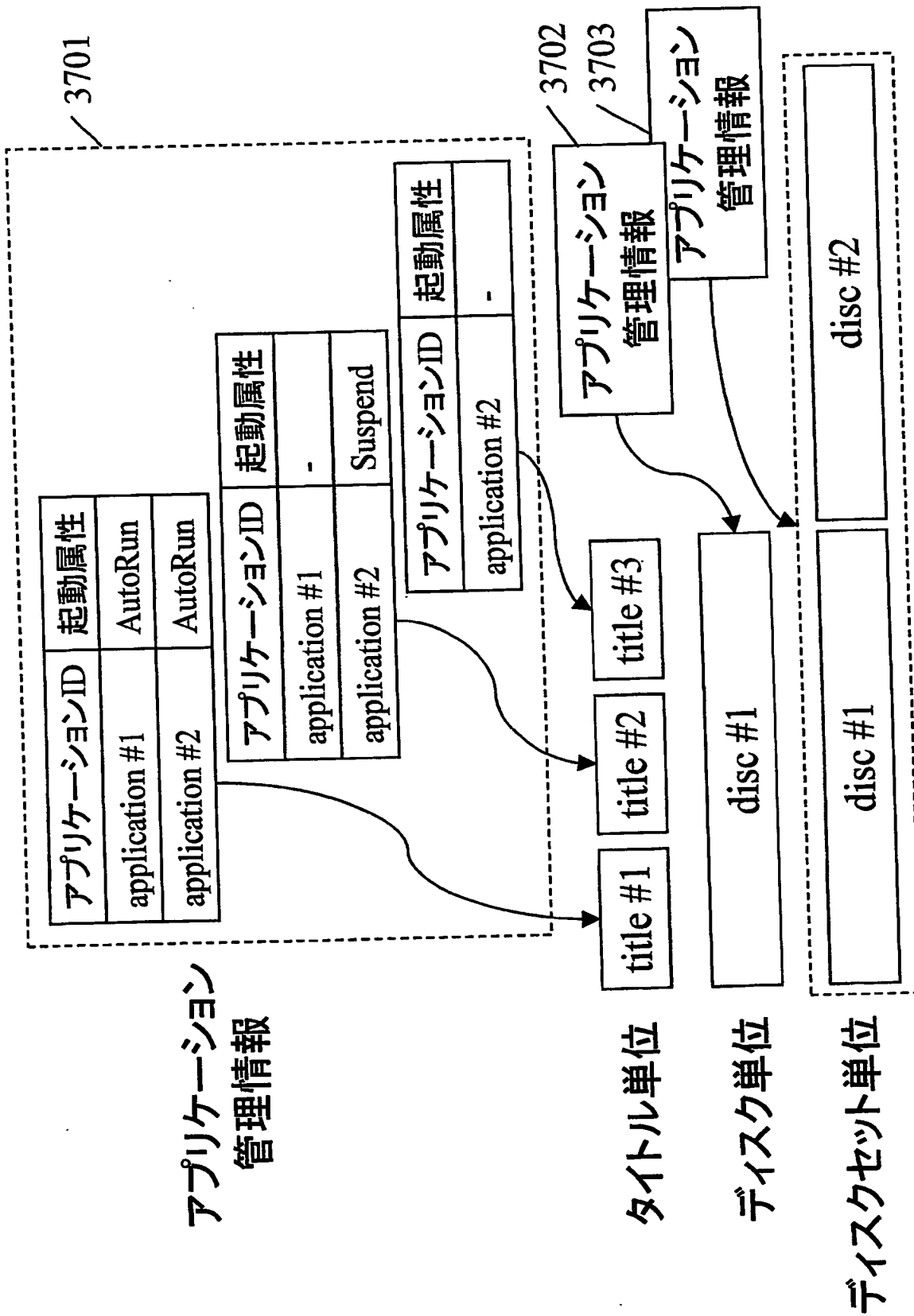
【図 36】

起動属性によるアプリケーション状態の変化

直前のタイトル における アプリケーション の状態	起動属性		
	(指定なし)	AutoRun	Suspend
	起動していない	アプリケーションを起動する	-
	起動している	-	サスペンドする
サスペンドしている	レジュームする	レジュームする	-

“-” : 何もせず、前の状態を継続する

【図 37】



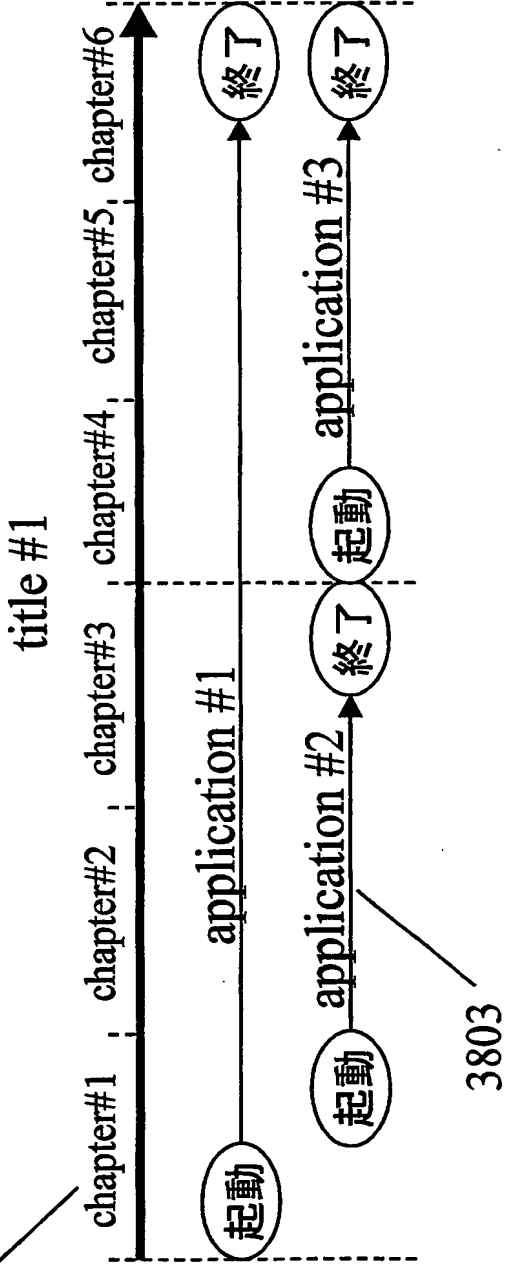
【図 3 8】

アプリケーション管理情報

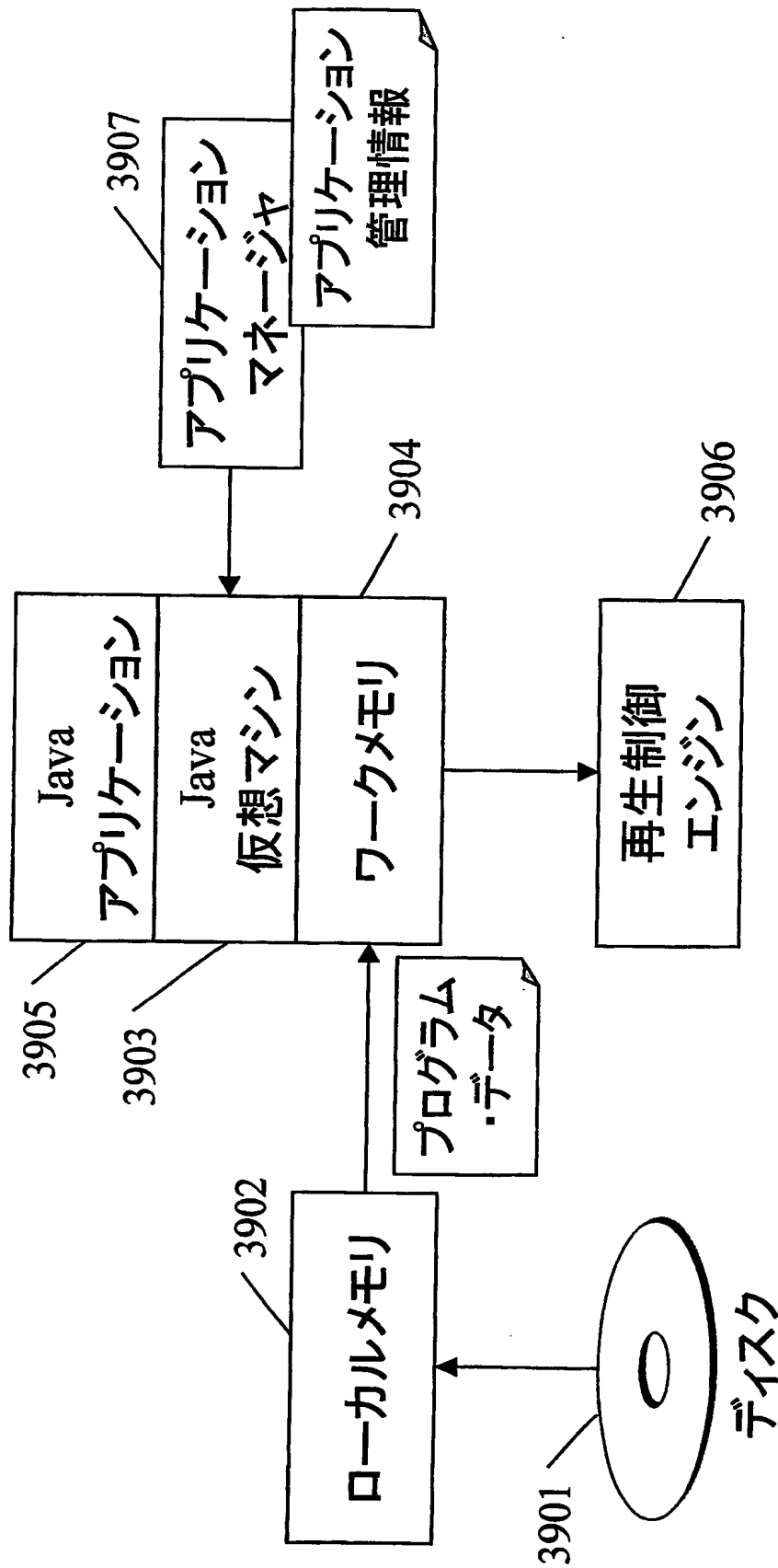
有効期間	アプリケーションID	起動属性
title #1	application #1	AutoRun
title #1 : chapter #1 - #3	application #2	-
title #1 : chapter #4 - #6	application #3	AutoRun

3802

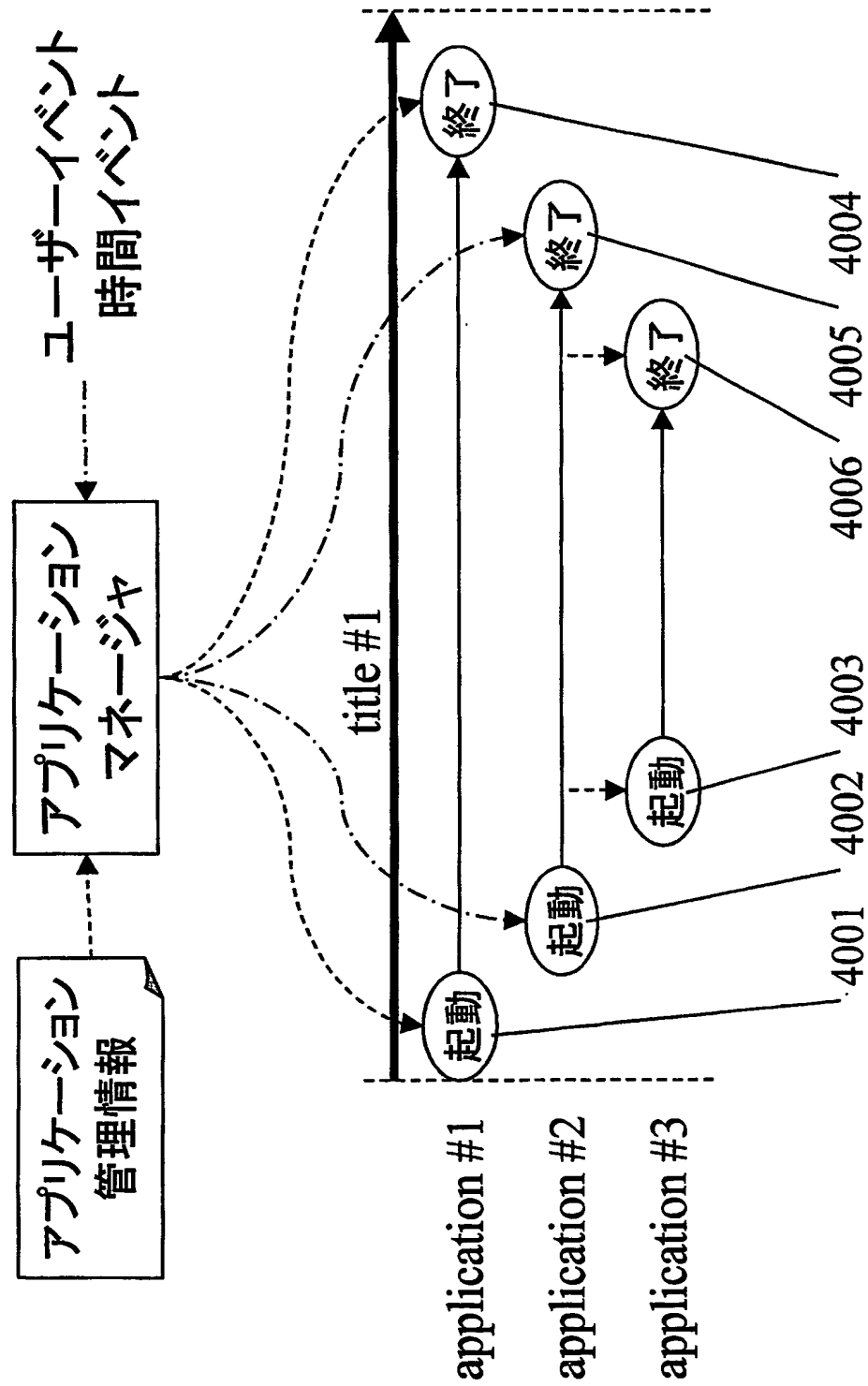
3801



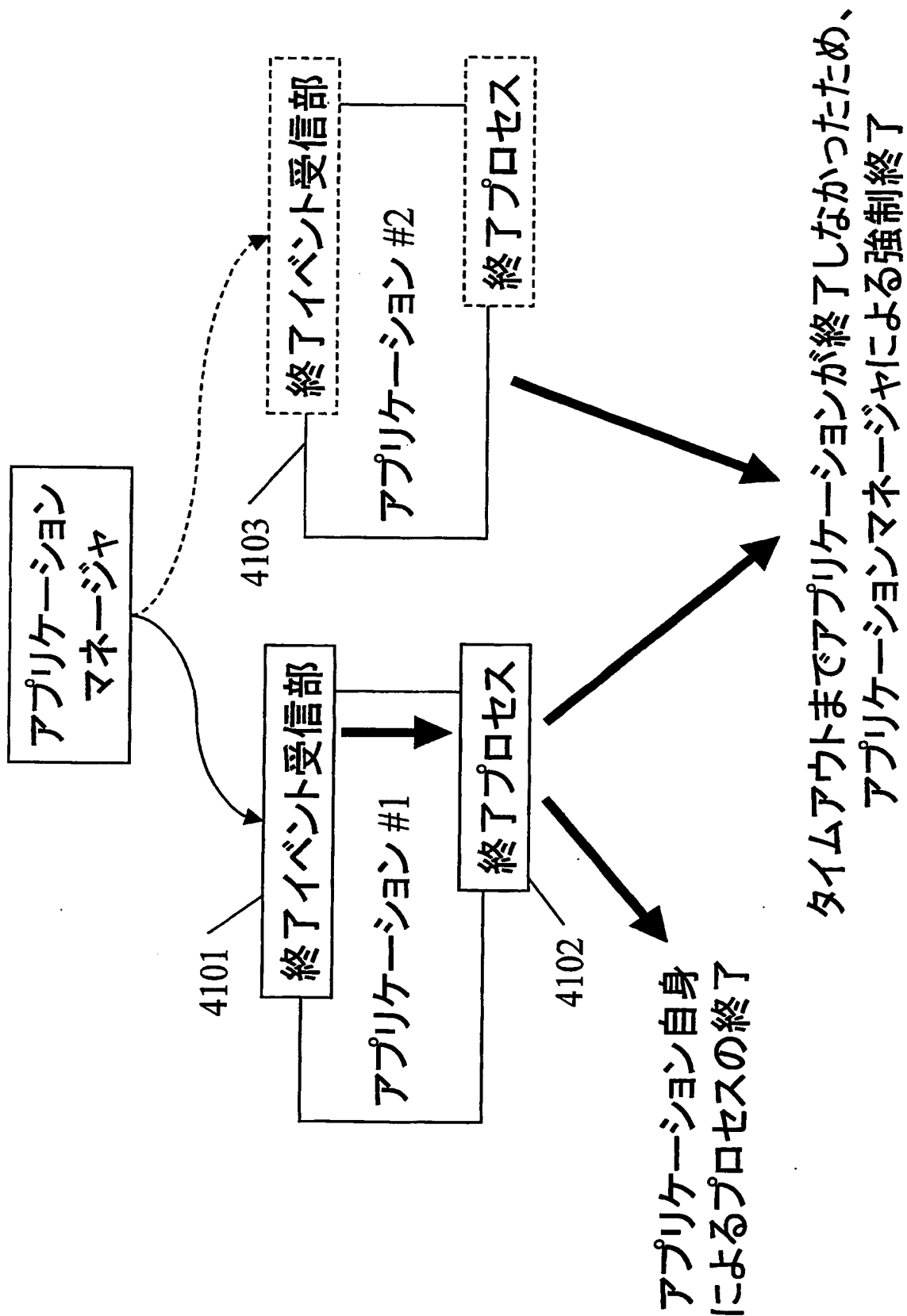
【図 39】



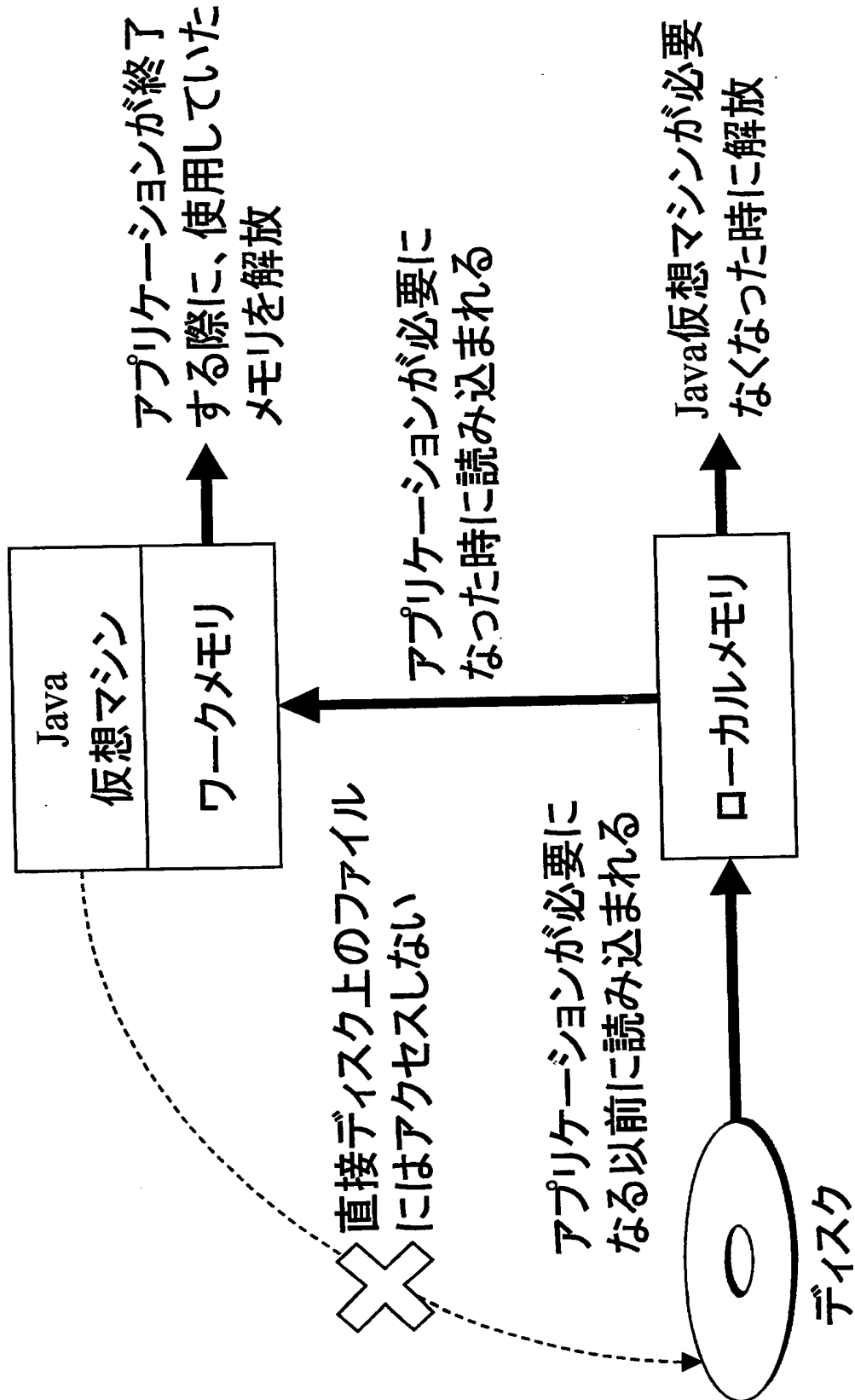
【図 40】



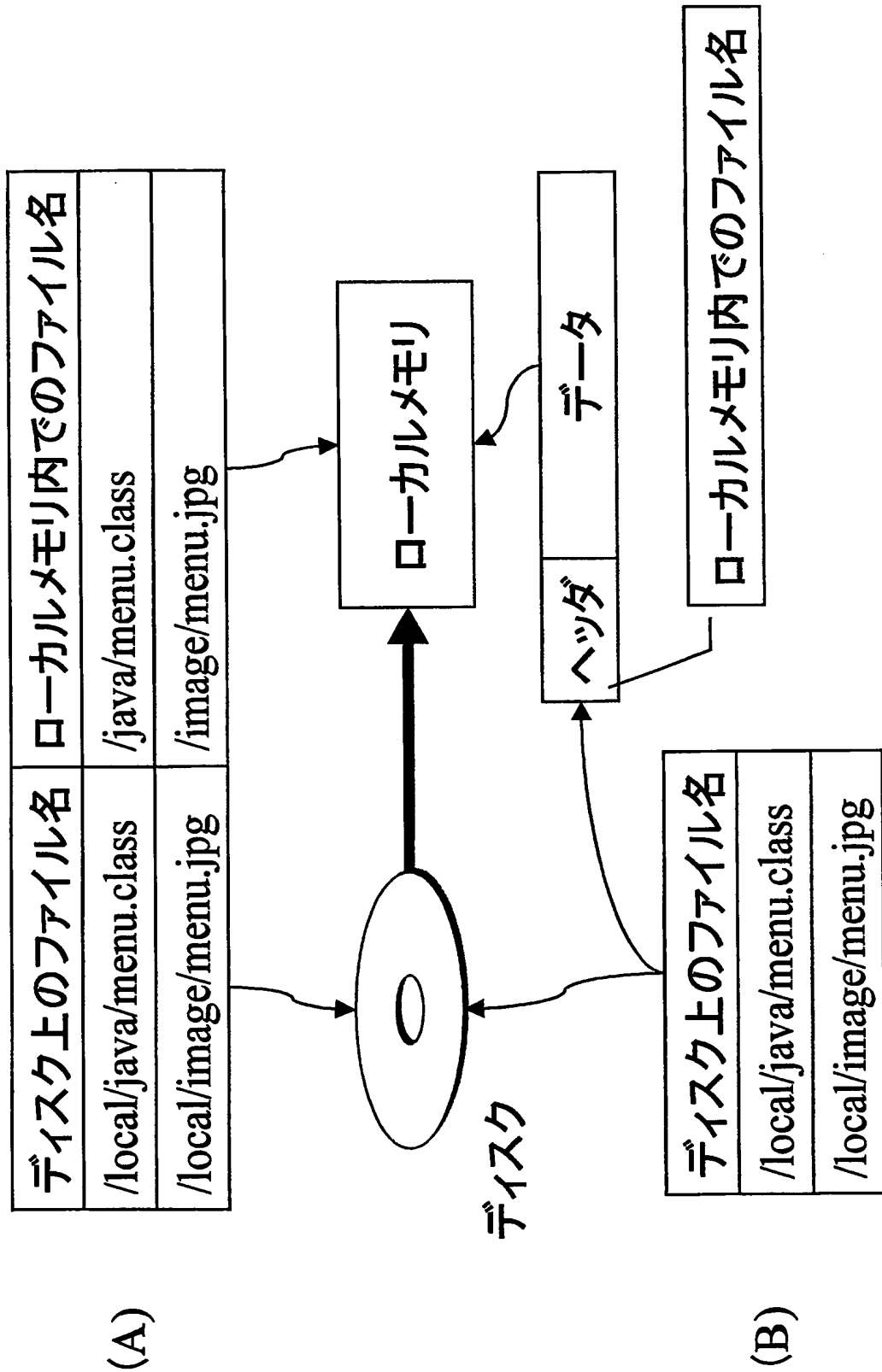
【図 4 1】



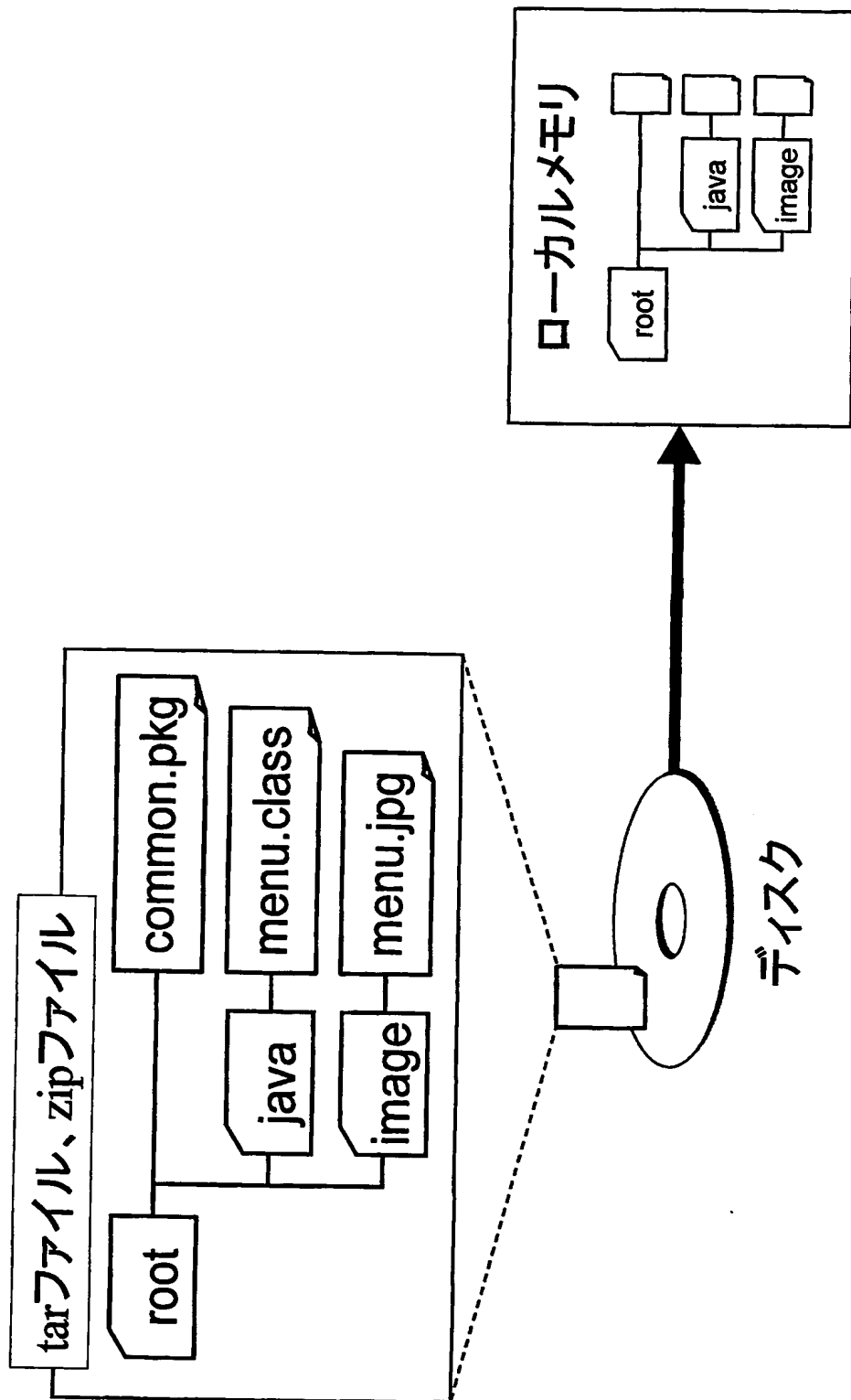
【図 42】



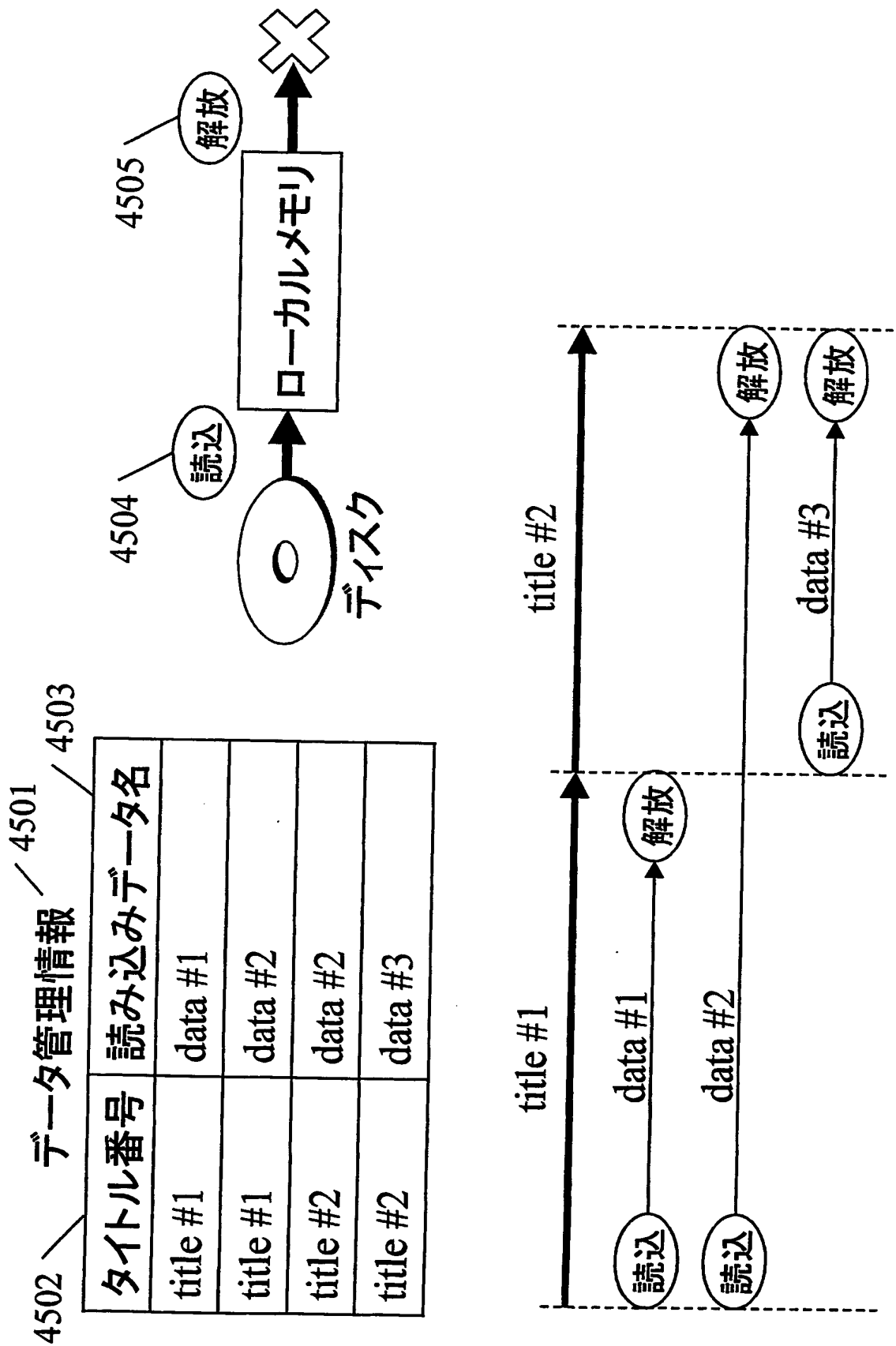
【図 43】



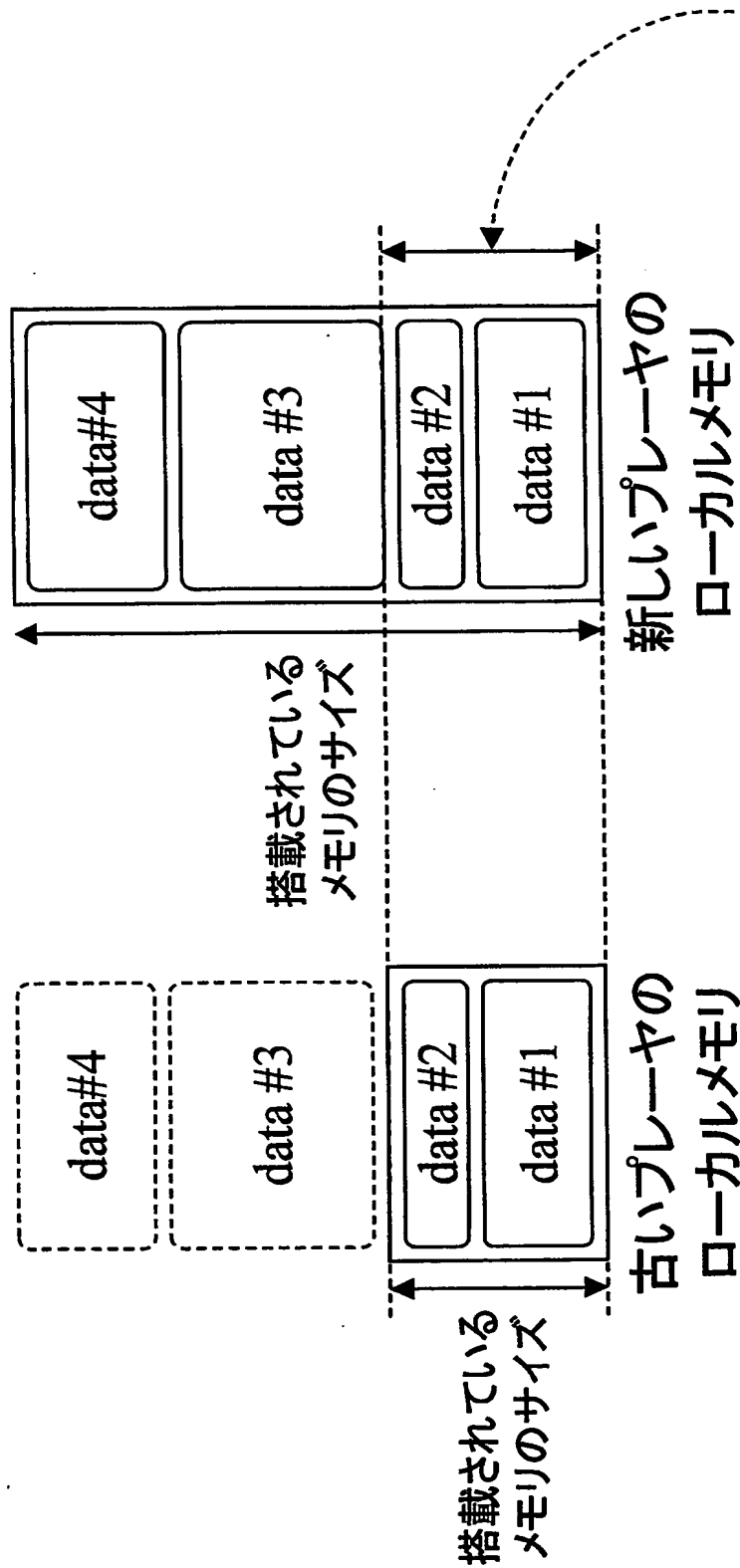
【図 44】



【図 45】

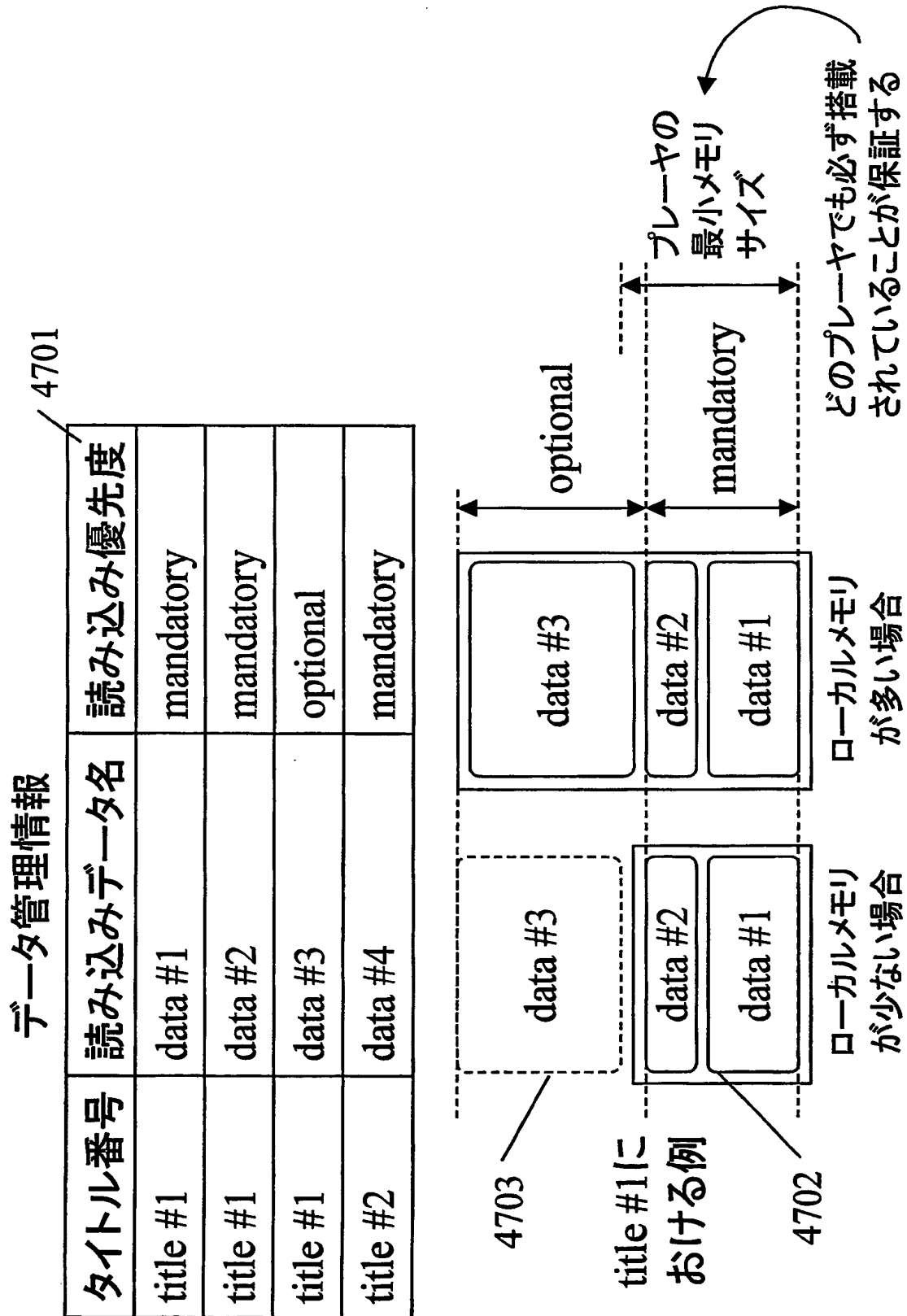


【図 46】

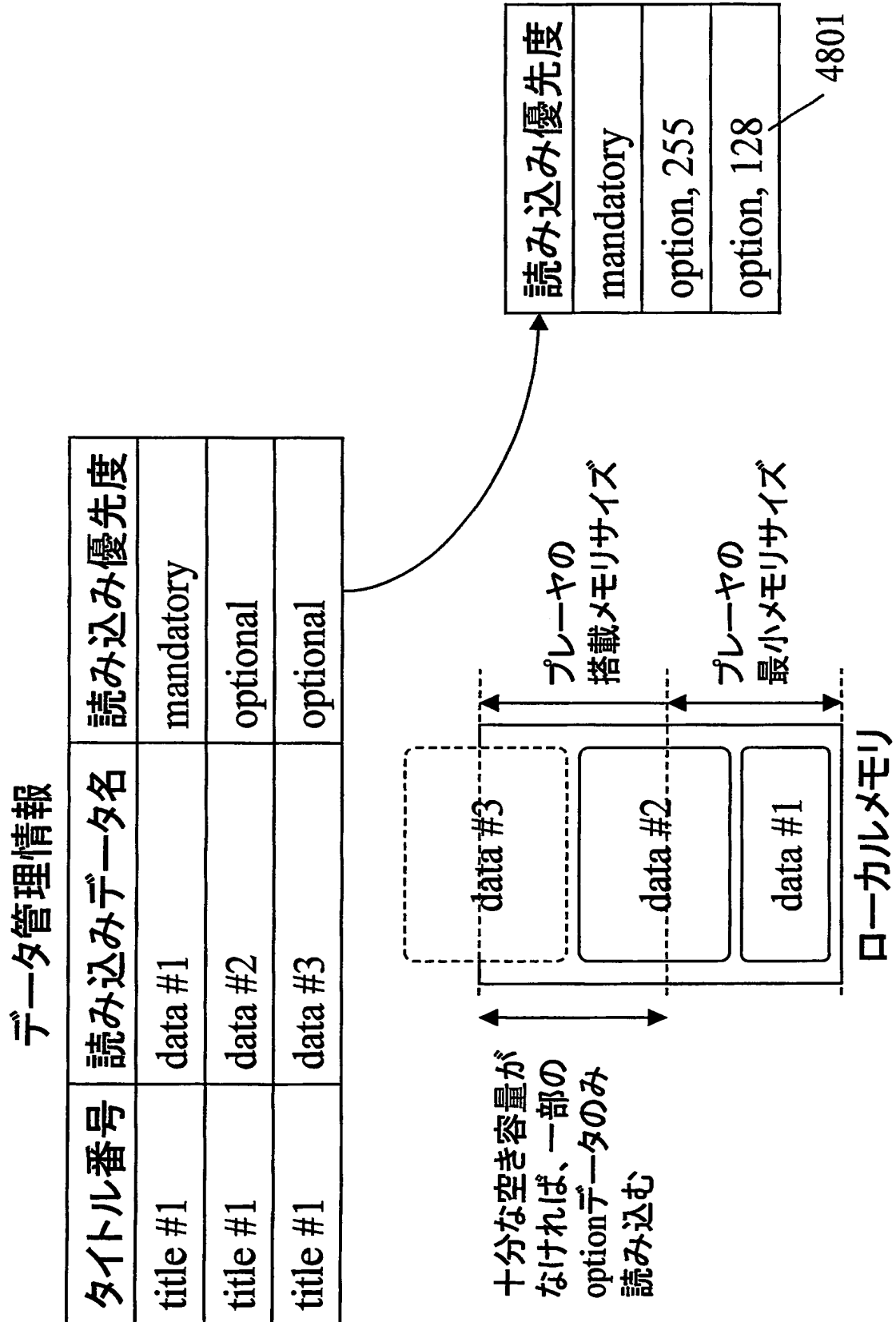


全てのプレーヤーで動作することを保証するには、
アプリケーションが使用するメモリサイズを
古いプレーヤーに合わせなければならない

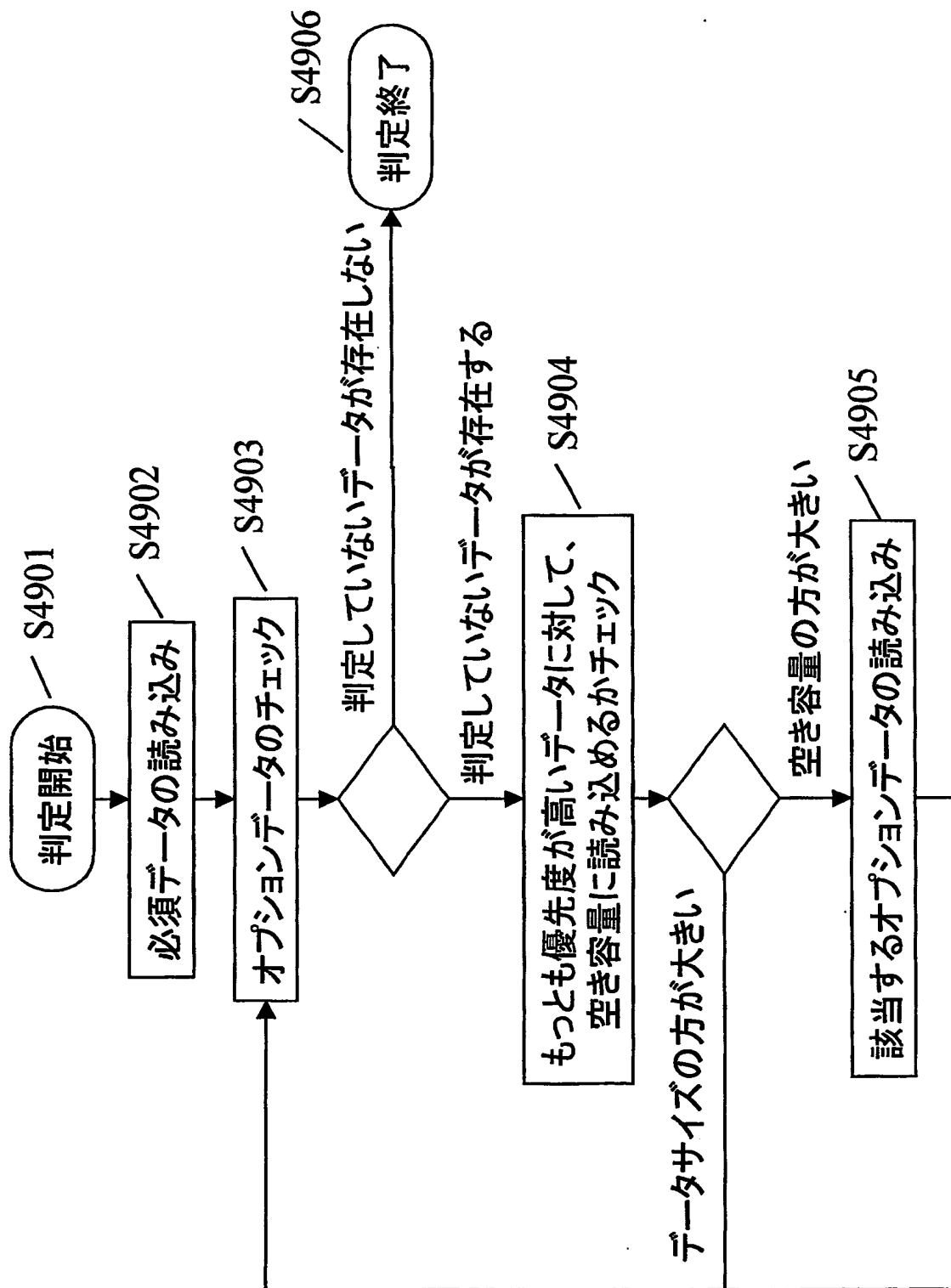
【図 47】



【図 48】



【図 49】



【図 50】

アプリケーション・データ管理情報

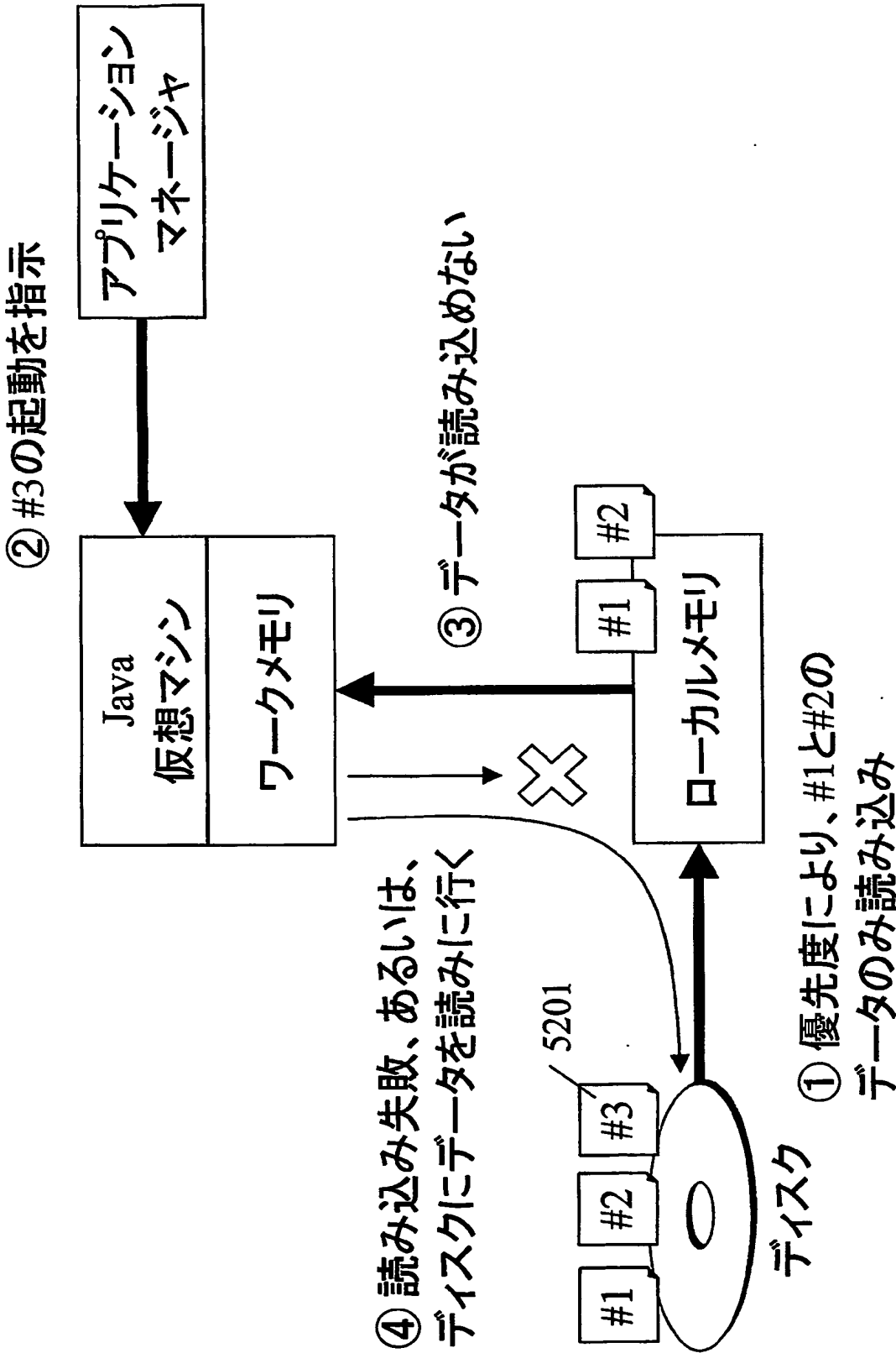
タイトル番号	アプリケーションID	起動属性	データ	優先度
title #1	application #1	AutoRun	00001.zip	mandatory
title #1	application #2	-	00002.zip	optional
title #2	application #3	-	00003.zip	optionnal

【図 5 1】

アプリケーション・データ管理情報 / 5101

タイトル番号	アプリケーションID	起動属性	データ
title #1	application #1	AutoRun (mandatory)	00001.zip
title #1:chapter #1-#3	application #2	AutoRun	00002.zip
title #1	application #3	-	00003.zip
title #1:chapter #2-#4	application #4	-	00004.zip

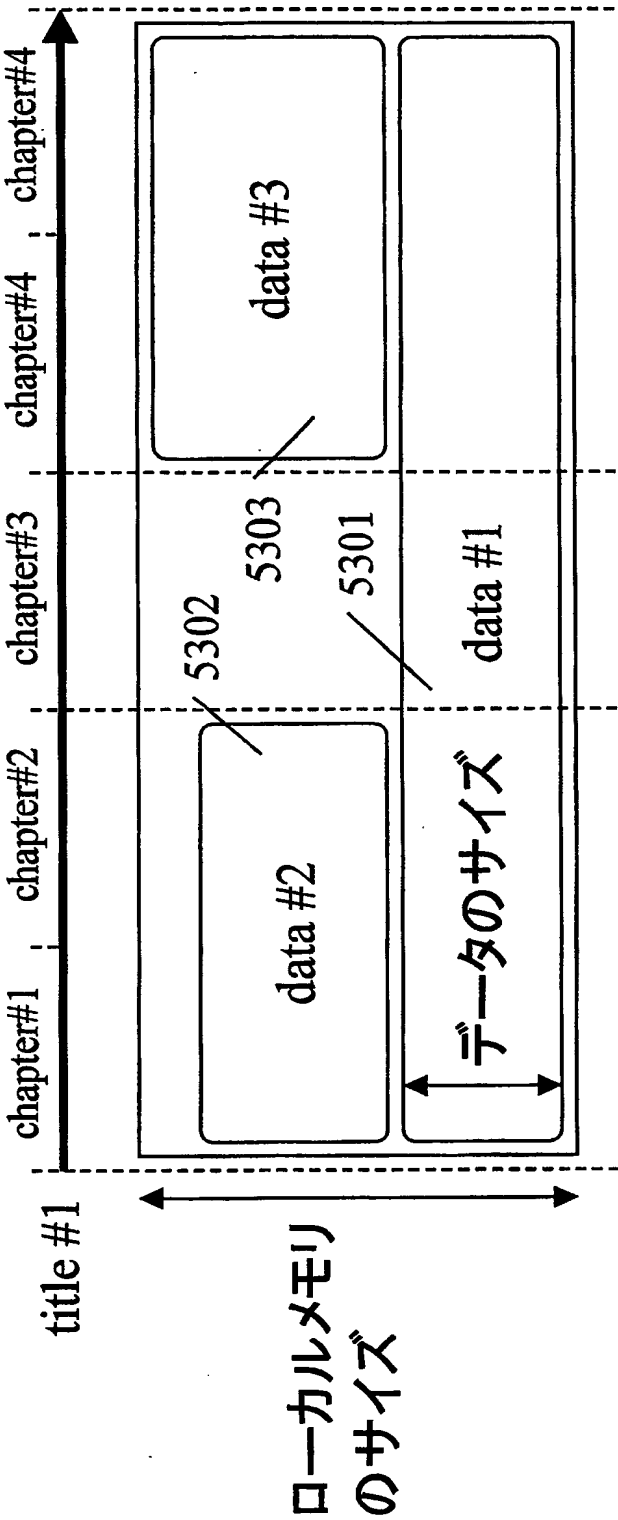
【図 5 2】



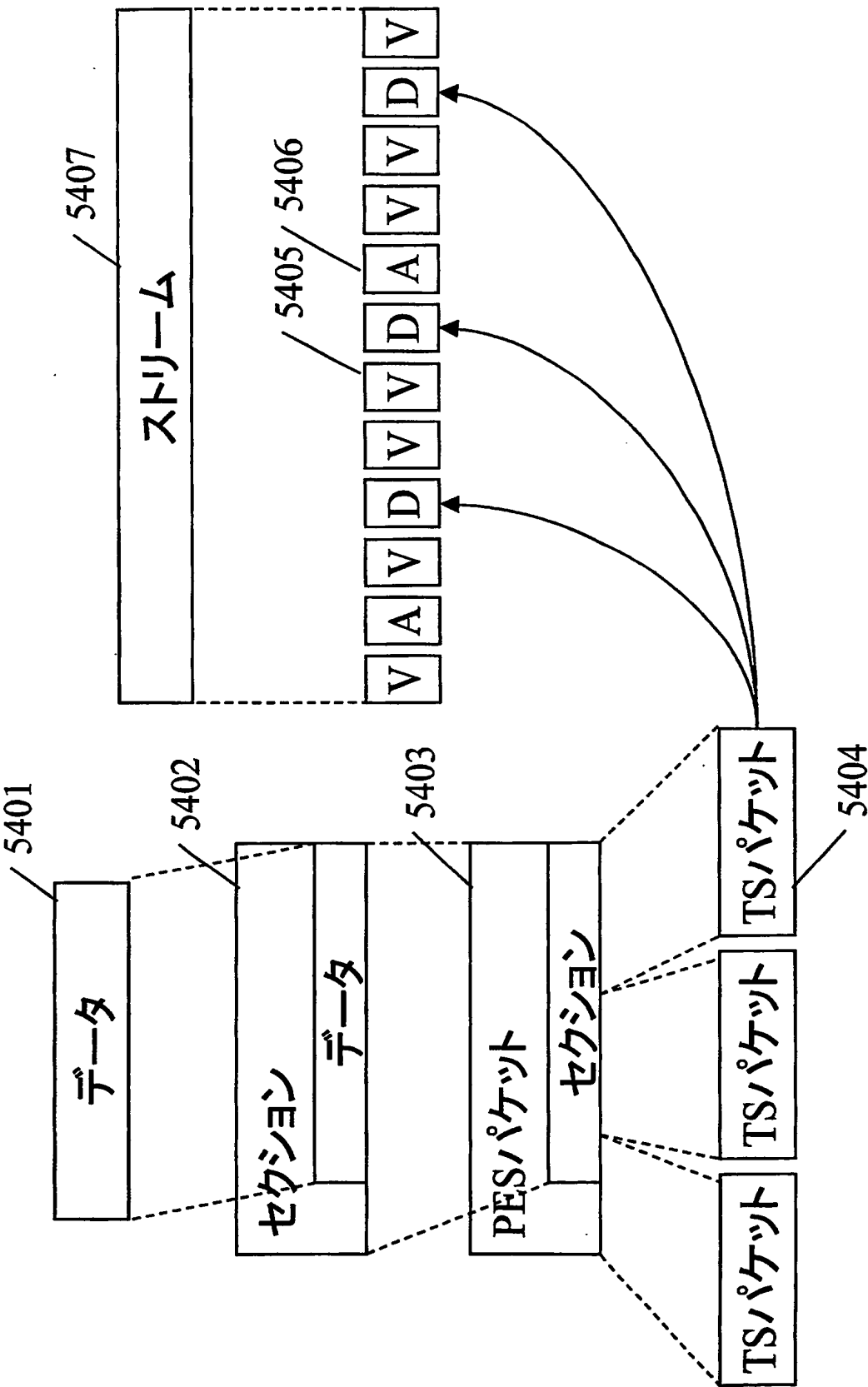
【図 5 3】

アプリケーション・データ管理情報

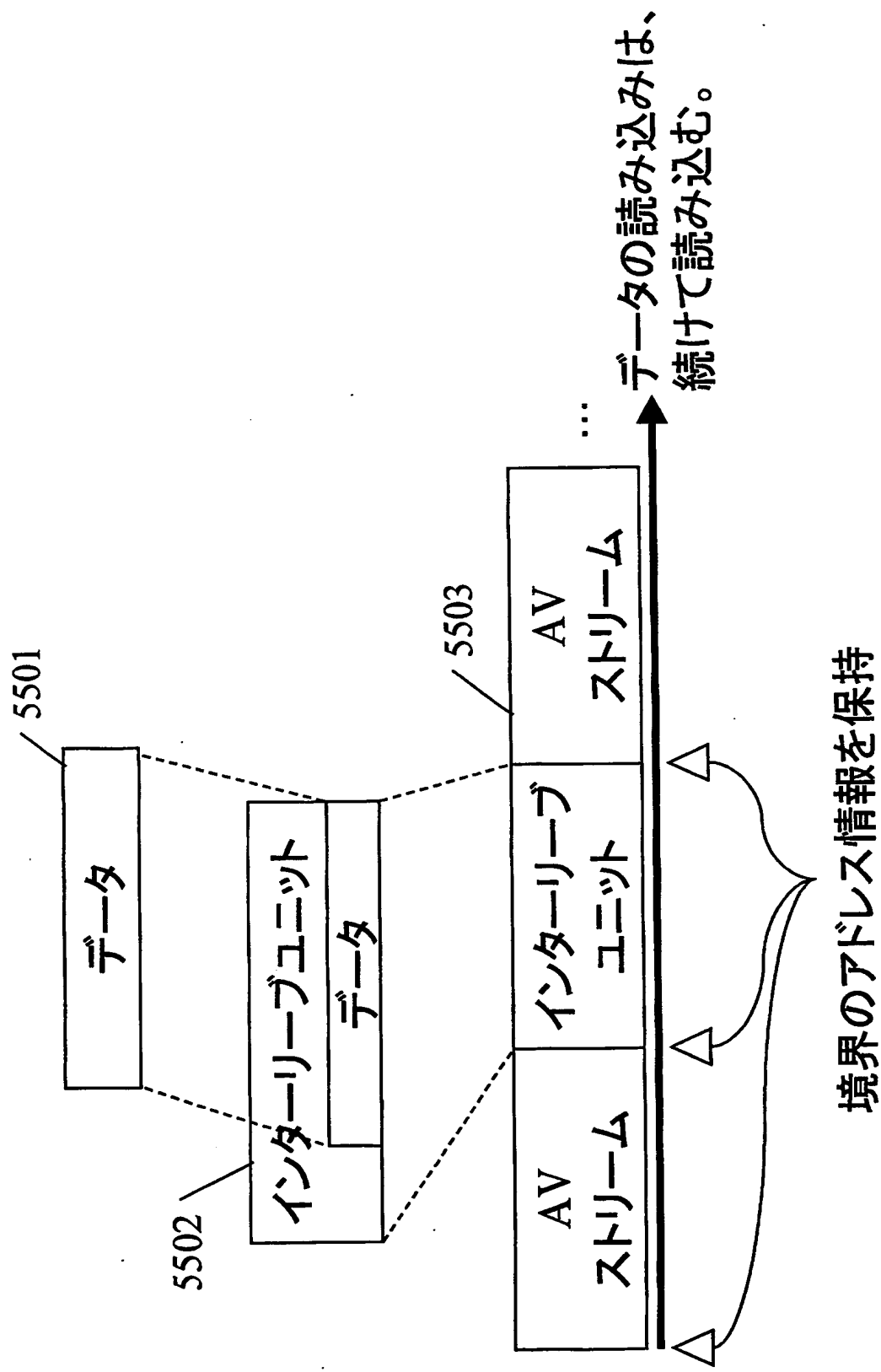
タイトル番号	アプリケーションID	起動属性	データ	優先度
title #1	application #1	AutoRun	data #1	mandatory
title #1:chapter #1-#2	application #2	AutoRun	data #2	optional
title #1:chapter #4-#5	application #3	AutoRun	data #3	optional



【図 5 4】



【図 55】



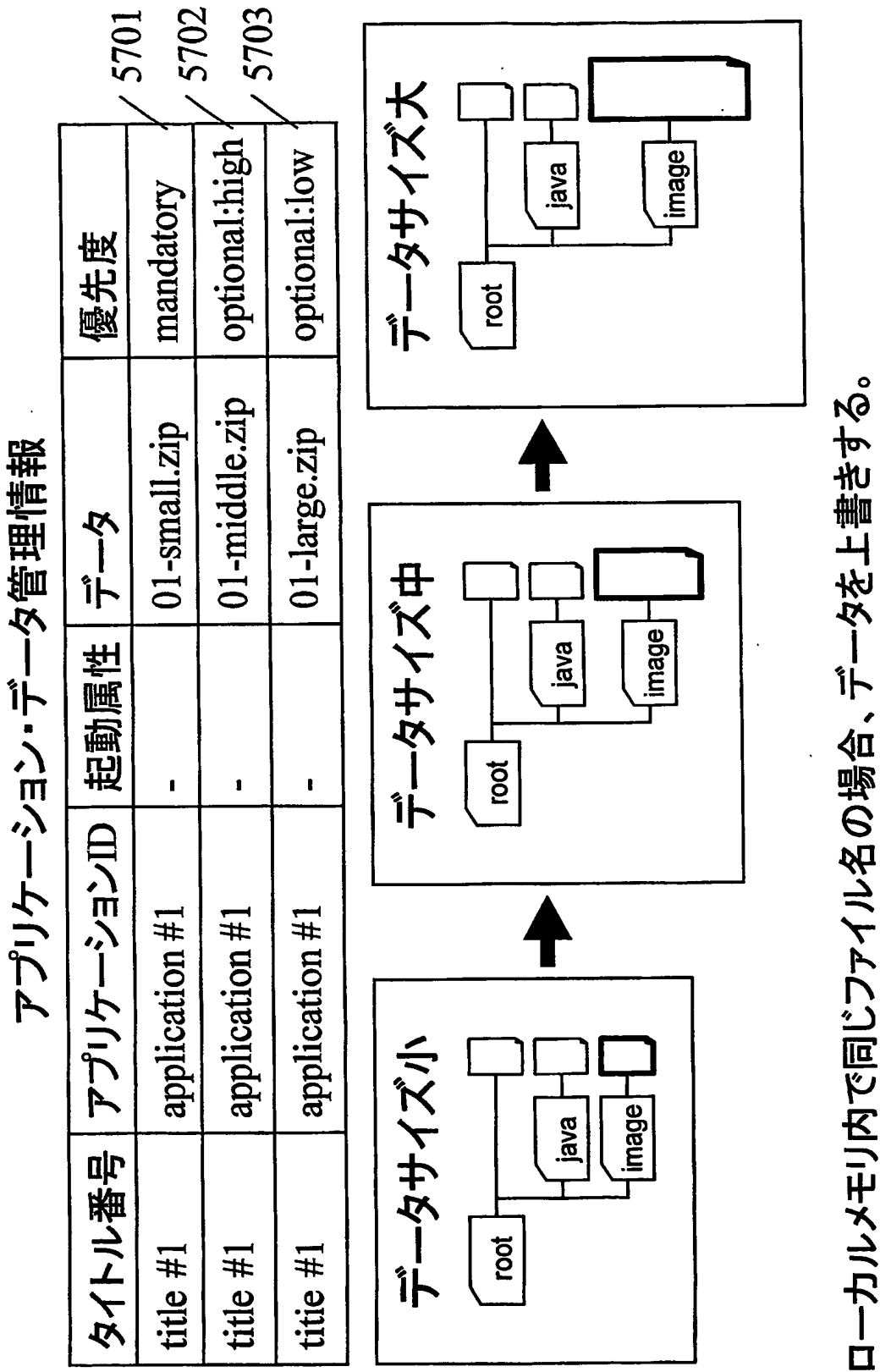
【図 5 6】

アプリケーション・データ管理情報

タイトル番号	アプリケーションID	起動属性	データ	優先度
title #1	application #1	AutoRun	01.zip	mandatory
title #1:chapter #1-#2	application #2	AutoRun	02.zip	mandatory
titie #1:chapter #4-#5	application #3	AutoRun	5601 ストリーム中	mandatory
titie #1:chapter #4-#5	application #3	AutoRun	5602 03.zip	optional

同じデータ

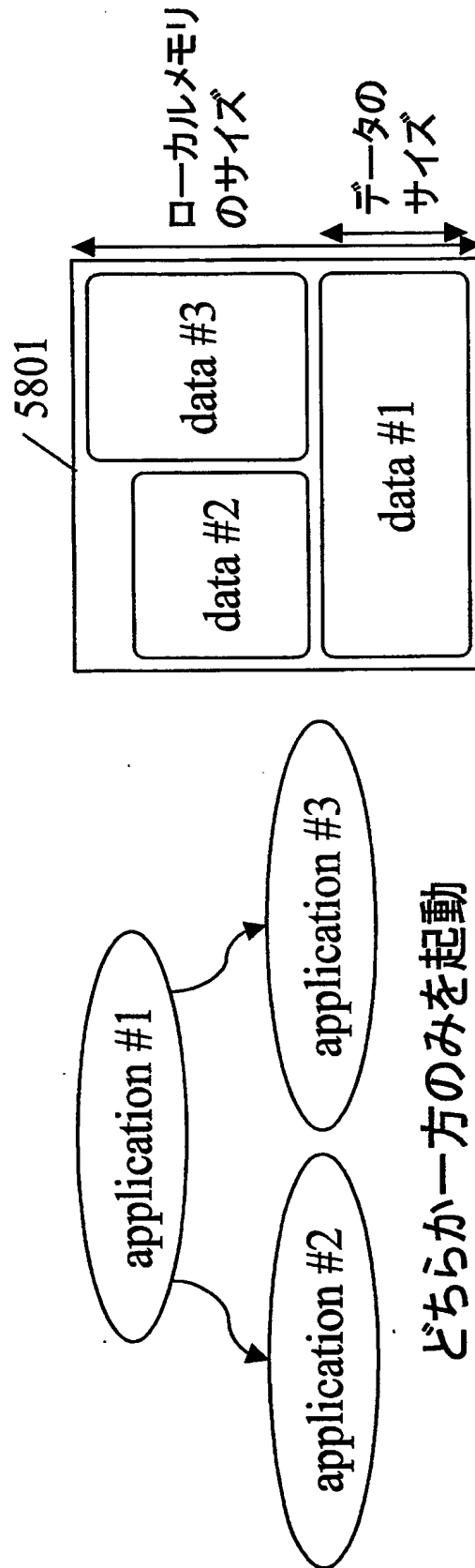
【図 5 7】



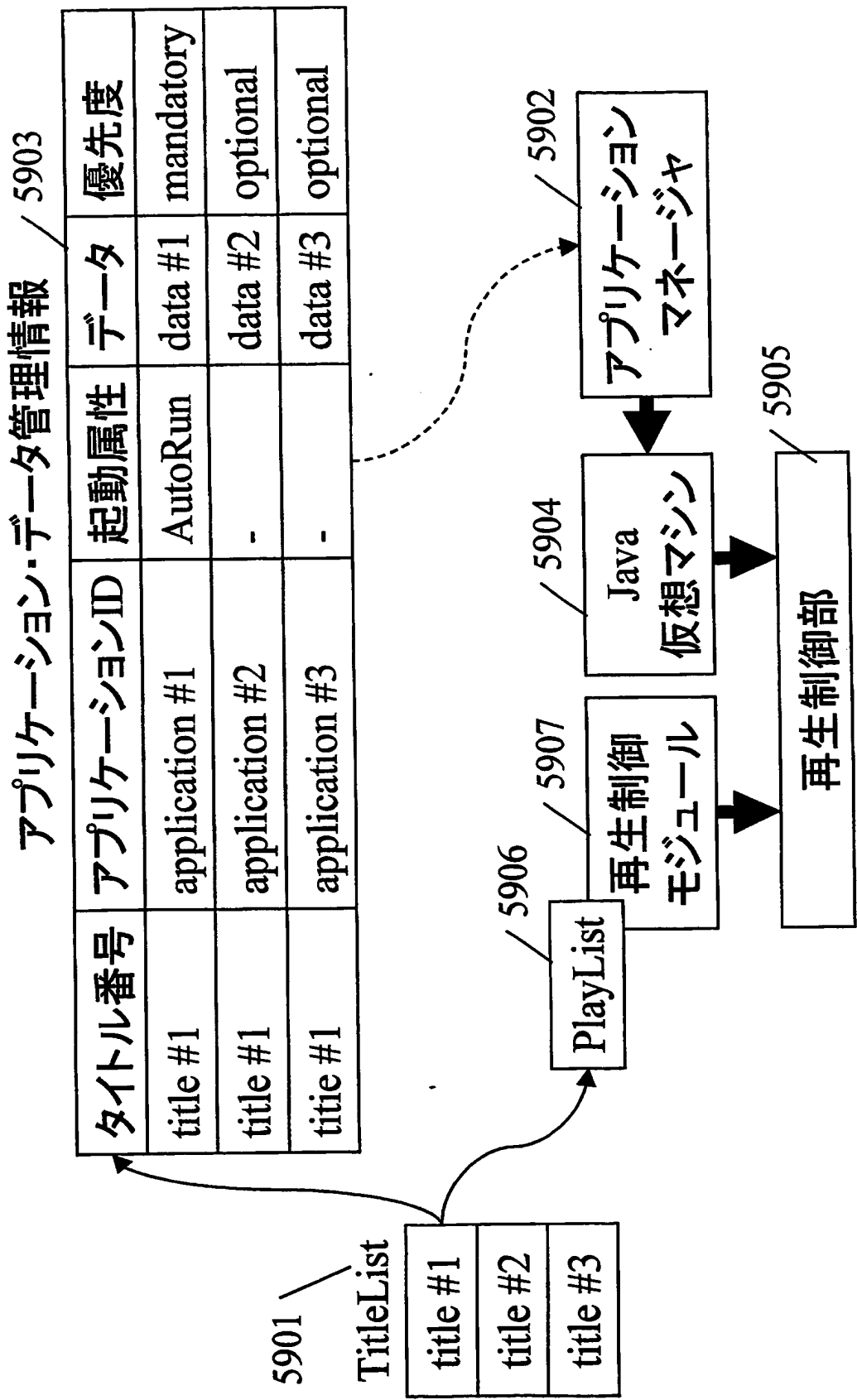
【図 58】

アプリケーション・データ管理情報

タイトル番号	アプリケーションID	起動属性	データ	優先度	排他グループ
title #1	application #1	AutoRun	01.zip	mandatory	-
title #1	application #2	-	02.zip	optional	group #1
title #1	application #3	-	03.zip	optional	group #1



【図 59】



【書類名】 要約書**【要約】**

【課題】 映像データ用BD-ROMで新たにプログラミング環境を導入するにあたって、タイトル等の管理単位内で動作するアプリケーションを容易に管理でき、管理単位外に影響を及ぼさない仕組みを提供する。

【解決手段】 情報記録媒体には、管理単位としてアプリケーション起動、終了、データ読み込み及び解放タイミングを管理する管理情報が記録されており、再生装置は、管理情報に基づいて制御を行うアプリケーションマネージャを備える。管理情報としてデータ読み込み優先度を用いることにより、メモリサイズに応じて読み込むべきデータサイズを調整することができる。

【選択図】 図50

特願 2 0 0 3 - 3 5 2 9 1 3

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 8 2 1]

1. 変更年月日	1 9 9 0 年 8 月 2 8 日
[変更理由]	新規登録
住 所	大阪府門真市大字門真 1 0 0 6 番地
氏 名	松下電器産業株式会社

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.